



GI-Edition

Lecture Notes in Informatics

Ira Diethelm (Hrsg.)

Informatische Bildung zum Verstehen und Gestalten der digitalen Welt

17. GI-Fachtagung Informatik und Schule

**13.–15. September 2017
Oldenburg**



Ira Diethelm (Hrsg.)

**Informatische Bildung zum Verstehen und
Gestalten der digitalen Welt**

17. GI-Fachtagung Informatik und Schule

**13.–15. September 2017
Oldenburg**

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Proceedings

Series of the Gesellschaft für Informatik (GI)

Volume P-274

ISBN 978-3-88579-668-8

ISSN 1617-5468

Volume Editor

Prof. Dr. Ira Diethelm

Carl von Ossietzky Universität Oldenburg

Fakultät II – Didaktik der Informatik

26111 Oldenburg, Deutschland

ira.diethelm@uni-oldenburg.de

Series Editorial Board

Heinrich C. Mayr, Alpen-Adria-Universität Klagenfurt, Austria

(Chairman, mayr@ifit.uni-klu.ac.at)

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, Infineon, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Andreas Thor, HFT Leipzig, Germany

Michael Goedicke, Universität Duisburg-Essen, Germany

Ralf Hofestädt, Universität Bielefeld, Germany

Michael Koch, Universität der Bundeswehr München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Thomas Roth-Berghofer, University of West London, Great Britain

Peter Sanders, Karlsruher Institut für Technologie (KIT), Germany

Torsten Brinda, Universität Duisburg-Essen, Germany

Ingo Timm, Universität Trier, Germany

Karin Vosseberg, Hochschule Bremerhaven, Germany

Maria Wimmer, Universität Koblenz-Landau, Germany

Dissertations

Steffen Hölldobler, Technische Universität Dresden, Germany

Thematics

Andreas Oberweis, Karlsruher Institut für Technologie (KIT), Germany

© Gesellschaft für Informatik, Bonn 2017

printed by Köllen Druck+Verlag GmbH, Bonn



This book is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

Vorwort

Digitalisierung durchdringt immer mehr Lebensbereiche, daher muss die Bildung darauf reagieren ... „Oh nein, nicht schon wieder!“, denken Sie vielleicht an dieser Stelle, und mir geht es ähnlich. Der Grund dafür ist aber erfreulich: Es verging in den letzten zwei Jahren kaum eine Woche, in der nicht eine politisch motivierte Tagung oder ein Feuilleton-Artikel sich dieses Themas in Bezug auf Bildung annahm. Sogar in Talkshows wurde das Thema schon hitzig und lautstark diskutiert. Immer mehr Personen aus den unterschiedlichsten Bereichen befassen sich jetzt neu mit den Themen Digitalisierung und Bildung bzw. Informatik und Schule.

Wir, die Aktiven der Fachdidaktik Informatik, die diese Tagung gestalten und ausrichten, durchlaufen daher gerade die Diskussionen der letzten 30 Jahre erneut und aufgrund des Fortschritts in den Kommunikationstechnologien auch viel schneller, intensiver und öffentlicher. Schon im „Gesamtkonzept für die informationstechnische Bildung“ der Bund-Länder-Kommission von 1987 findet sich die Eingangserkenntnis: „Neue Informations- und Kommunikationstechniken haben sich in den letzten Jahren zu Schlüsseltechniken entwickelt, die in fast alle Bereiche [...] Einzug gehalten haben; sie haben auch im Bildungswesen und in unserer täglichen Umwelt Fuß gefaßt.“

Schon damals fragte man sich, welche Bildung für diese Veränderungen nötig ist, ob es ein Schulfach Informatik geben sollte, wie dieses oder ein Minimalprogramm „24 Lektionen Informatik“ aufgebaut sein sollten, was allgemeinbildend an der Informatik ist, wie das Verhältnis zum Einsatz von Computern als Werkzeug und Unterrichtsmedium ist, welchen Beitrag Informatik zur Medienkompetenz leistet, ab wann man mit Kindern Informatik machen kann und ob und ab wann Programmieren Pflicht sein sollte usw.

Seit 1976 gibt die Gesellschaft für Informatik e.V. in Form von Empfehlungen und Erklärungen Antworten auf diese Fragen, seit 1984 auch in den INFOS-Bänden. Dort findet man viele noch immer aktuelle Antworten. Zusammen bilden sie eine Art FAQ zur Informatik in der Schule. Jedem Einsteiger in das Thema seien diese Werke wärmstens empfohlen.

Schon im ersten INFOS-Band stand: „Die Kultusminister haben mehr oder weniger anspruchsvolle Programme zur Integration der Informatik und zur Rechnerausstattung der Schulen angekündigt.“ Auch inhaltlich sind die Überlegungen von damals der jüngsten Strategie der Kultusministerkonferenz zur Bildung in der digitalen Welt nicht unähnlich. Die Politik scheint sich zu wiederholen. Damals wie heute war klar: Es reicht nicht Computer in die Schulen zu stellen. Die Geschichte und auch die Lehrerbildungsforschung hat gezeigt, dass es auch nicht reicht, Unterrichtsmaterialien zu drucken und an die Schulen zu schicken, in der Hoffnung, dass die Lehrkräfte oder gar Schüler sich das schon irgendwie selbst aneignen, den Sinn verstehen und flächendeckend umsetzen.

Aber was reicht denn dann für die sogenannte „Digitale Bildung“, also die Bildung für die und in einer von der Digitalisierung geprägten Welt? Was tun? In den eingeladenen Vorträgen nehmen sich Jens Gallenbacher, Hilbert Meyer, Thomas Knaus und Stephan Noller dieser Frage aus verschiedenen Perspektiven an, kommen aber alle am selben Punkt

zusammen: Eine flächendeckende informatische Bildung ist zum Verstehen und Gestalten der digitalen Welt unerlässlich. Auch Hilbert Meyer, der sich hier erstmals mit dem Thema auseinandersetzt, spricht sich für ein Schulfach Informatik in den Sekundarstufen aus und Thomas Knaus betont, dass Medienbildung nicht genug sei und informatische Bildung zur Medienkompetenz dazugehöre.

Diese Erkenntnis ist für die meisten regelmäßigen INFOS-Teilnehmer nicht neu. Neu ist, dass sie sich auch außerhalb dieses Kreises weiter ausbreitet. Immer mehr Länder und Bundesländer führen das Schulfach Informatik (ab Sommer 2017 zuletzt Baden-Württemberg und Finnland), „Computing“ (England), den Lehrplan 21 „Informatik und Medien“ (Schweiz) oder etwas Vergleichbares verpflichtend ein oder haben so etwas schon seit fast 30 Jahren (Polen).

Wie vielseitig das Thema jenseits der politischen Dimension ist, zeigen die Beiträge der weiteren Autoren, an deren Anzahl man ebenfalls die Breite der Diskussion ablesen kann. Es wurden 98 Beiträge eingereicht, darunter auch viele, die sich mit der Informatik in der frühen Bildung beschäftigen. In der Regel wurde jeder Beitrag von 3 Reviewern anonym mithilfe von EasyChair begutachtet. Das Programmkomitee hat 47 Beiträge, teilweise mit Änderungen, akzeptiert. Darunter sind 13 wissenschaftliche Langbeiträge, 6 wissenschaftliche Kurzbeiträge, 13 Praxisbeiträge und 9 Workshops sowie 6 Poster. Daraus resultiert eine Annahmequote von 48%. Der Band enthält darüber hinaus vier eingeladene Keynotes, von denen zwei als Langfassungen und zwei mit Abstracts enthalten sind. Außerdem wurden einige in ihrer eingereichten Fassung abgelehnte Beiträge in stark gekürzter Form als Poster eingeladen. Alle Posterabstracts sind am Ende des Bandes alphabetisch nach Autoren gereiht. Die Reihung der Keynotes, Vorträge und Workshops folgt dem Tagungsprogramm.

Die „Informatik und Schule“ ist die größte deutschsprachige Konferenz zur Informatik in der Schule. So stammt die überwiegende Mehrheit der AutorInnen aus Deutschland. Aber der Band enthält auch Beiträge von Autoren aus Dänemark und der Schweiz.

Zur Durchführung der Tagung sind viele Hände und viele Unterstützer nötig. Ich möchte mich daher allen voran bei meinen Mitarbeitern der Abteilung Didaktik der Informatik ganz herzlich bedanken. Mein Dank gilt auch der Universität Oldenburg mit der Universitätsgesellschaft (UGO), dem An-Institut OFFIS und der Gesellschaft der Freunde und Förderer des OFFIS e.V. (GdFF) für die tatkräftige und finanzielle Unterstützung der Tagung. Ebenso danke ich dem Programmkomitee für die Mitarbeit bei dem Auswahlverfahren und insbesondere Torsten Otto und Arno Pasternak für die Begleitung des ganzen Prozesses. Darüber hinaus geht mein Dank an das Niedersächsische Landesinstitut für schulische Qualitätsentwicklung (NLQ) und die Fachgruppe GI-ibnb und natürlich an die vielen Autoren, ohne die diese Tagung gar nicht stattfinden würde.

Oldenburg, im August 2017

Ira Diethelm

Tagungsleitung

Gesamtleitung: Ira Diethelm, Universität Oldenburg
Leitung des Programmkomitees: Ira Diethelm, Universität Oldenburg
Torsten Otto, GI-Fachgruppe SH-HILL
Arno Pasternak, TU Dortmund
DoktorandInnensymposium: Nils Pancratz, Universität Oldenburg
Jörn Syrbe, Universität Oldenburg

Programmkomitee

Nadine Bergner	RWTH Aachen
Torsten Brinda	Universität Duisburg-Essen
Michael Brinkmeier	Universität Osnabrück
Ulrike Buchholz	Gymnasium Oedeme Lüneburg
Katrin Büttner	Mittelschule Heidenau (stellv. Sprecherin FA-IBS)
Ira Diethelm	Carl von Ossietzky Universität Oldenburg
Leonore Dietrich	Universität Heidelberg
Beat Döbeli Honegger	Pädagogische Hochschule Schwyz
Jens Gallenbacher	TU Darmstadt
Lutz Hellmig	Universität Rostock (Sprecher FA-IBS)
Tino Hempel	Richard-Wossidlo-Gymnasium
Henry Herper	Universität Magdeburg
Alexander Hug	Universität Koblenz-Landau
Ludger Humbert	Universität Wuppertal
Lennard Kerber	GI-Fachgruppe IBBB
Bernhard Koerber	LOG IN Verlag GmbH
Peter Micheuz	Alpen-Adria-Universität Klagenfurt
Simone Opel	Berufskolleg der Stadt Bottrop
Torsten Otto	GI-Fachgruppe SH-HILL
Arno Pasternak	TU Dortmund
Wolfgang Pohl	BWINF / Bundesweite Informatikwettbewerbe
Juergen Poloczek	Studienseminar Oberursel
Mareen Przybylla	Universität Potsdam
Ralf Romeike	Friedrich-Alexander-Universität Erlangen-Nürnberg
Gerhard Röhner	Studienseminar für Gymnasien Darmstadt
Carsten Schulte	Universität Paderborn
Andreas Schwill	Universität Potsdam
Kerstin Strecker	Universität Göttingen
Michael Weigend	WWU Münster

Organisationsteam

Christos Chytas
Mareike Daeglau
Wibke Duwe
Anatolij Fandrich
Claudia Hildebrandt
Nils Pancratz
Tobias Stuckenberg
Jörn Syrbe
Manuela Wüstefeld

Carl von Ossietzky Universität Oldenburg
Carl von Ossietzky Universität Oldenburg
Carl von Ossietzky Universität Oldenburg
OFFIS e.V.
Carl von Ossietzky Universität Oldenburg
Carl von Ossietzky Universität Oldenburg
Carl von Ossietzky Universität Oldenburg
Carl von Ossietzky Universität Oldenburg
Carl von Ossietzky Universität Oldenburg

Lektorat

Nina Schneider

Carl von Ossietzky Universität Oldenburg

Proceedings-Support

Anatolij Fandrich
Nils Pancratz
Jörn Syrbe

OFFIS e.V.
Carl von Ossietzky Universität Oldenburg
Carl von Ossietzky Universität Oldenburg

Inhaltsverzeichnis

Eingeladene Vorträge

Jens Gallenbacher <i>Allgemeinbildung in der digitalen, gestalteten Lebenswelt</i>	19
Hilbert Meyer <i>Unterrichtsqualität in der digitalen Welt</i>	29
Thomas Knaus <i>Verstehen – Vernetzen – Verantworten – Warum Medienbildung und informatische Bildung uns alle angehen und wir sie gemeinsam weiterentwickeln sollten</i>	31
Stephan Noller <i>Calliope mini – Informatik zum Anfassen für GrundschülerInnen</i>	49

Vorträge

Nadine Bergner, Hilde Köster, Johannes Magenheimer, Kathrin Müller, Ralf Romeike, Ulrik Schroeder, Carsten Schulte <i>Zieldimensionen für frühe informatische Bildung im Kindergarten und in der Grundschule</i>	53
Nadine Bergner, Sandra-Jasmin Petrut, Ulrik Schroeder <i>Was Grundschulkindern über Informatik wissen und was sie wissen wollen</i>	63
Katharina Wendlandt, Matthias Wendlandt, Sabrina Hoffmann <i>Algorithmisieren im Grundschulalter</i>	73
Alexander Best <i>Bild der Informatik von Grundschullehrpersonen. Erste Ergebnisse aus qualitativen Einzelfallstudien</i>	83

Andreas Dengel, Ute Heuer <i>Aufbau des Internets: Vorstellungsbilder angehender Lehrkräfte</i>	87
Beat Döbeli Honegger, Michael Hielscher <i>Vom Lehrplan zur Lehrerinnenbildung - Erste Erfahrungen mit obligatorischer Informatikdidaktik für angehende Schweizer Primarlehrerinnen . . .</i>	97
Kathrin Müller, Carsten Schulte <i>Ein Modell zur Analyse von Vorstellungen über Roboter und ihrer Funktionsweise</i>	109
Torsten Brinda, Friederike Braun <i>Schülervorstellungen im Zusammenhang mit Smartphones</i>	119
Yannick Schneider, Andreas Mühling <i>Das Konzept Nature of Computer Science</i>	123
Dorothee Müller <i>Berufswahl Informatiklehrkraft</i>	127
Claudia Hildebrandt <i>Mit dem Glauben Berge versetzen ... - Die Selbstwirksamkeitserwartung von Informatiklehrkräften</i>	137
Nico Steinbach, Eckart Zitzler <i>Ein gamebasierter Ansatz zum Programmierunterricht in der Lehrerinnen- und Lehrerbildung</i>	147
Andreas Grillenberger, Ralf Romeike <i>Empirische Ermittlung der Schlüsselkonzepte des Fachgebiets Datenmanagement</i>	157
Alexander Hug, Rüdiger Grimm <i>Entwicklung eines Datenschutzkompetenzmodells</i>	167
Klaus-Tycho Förster <i>Vom Flaggenalphabet zur Vorratsdatenspeicherung: Schülerinnen und Schüler als Multiplikatoren technischer Aspekte der digitalen Welt</i>	171
Jonathan Otto, Michel Welke, Leonard Diekmann, Maria Knobelsdorf <i>Hamburg 2050, Land unter? – Eine „Informatik im Kontext“-Reihe zu Klimamodellierung mit Geographischen Informationssystemen</i>	181

Esther Alzate Romero, Leonore Dietrich <i>Musikprogrammierung mit Sonic Pi. Entwicklung und Untersuchung einer gendersensiblen Unterrichtseinheit zum Programmieren in der Sekundarstufe I.</i>	191
Dominik Heun <i>Entwicklung eines HTML-Editors unter didaktischen Gesichtspunkten . . .</i>	201
Johanna Borsch, Marco Thomas <i>Informatische Bildung in der Sekundarstufe I an nordrhein-westfälischen Schulen</i>	211
Lars Prädell, Gerlinde Schreiber <i>Hochschule als außerschulischer Lernort für Schülerinnen und Schüler: Ein pragmatisches Angebot</i>	221
Dieter Engbring <i>Aller Anfang ist schwer! Wie gelingt der Einstieg in den Informatikunterricht?</i>	227
Nadine Bergner, Thimeo Leonhardt, Ulrik Schroeder <i>Fünf Argumente für einen grafischen ProgrammierEinstieg - eine Studie über vier Jahrgangsstufen</i>	237
Johannes Fischer, Arno Pasternak <i>Modularisierung im Informatikunterricht aus lernpsychologischer Perspektive</i>	247
Mareen Przybylla, Ralf Romeike <i>Von Eingebetteten Systemen zu Physical Computing: Grundlagen für Informatikunterricht in der digitalen Welt</i>	257
Tobias Jördens, Jens Gallenbacher <i>Quadrologik - Modellbildung und Modularisierung auf Basis von Rechner-technik</i>	267
Marlene Lindner, Sandra Schulz, Niels Pinkwart <i>Integration des Erwerbs von Basiskonzepten der Informatik in den mathematisch-naturwissenschaftlichen Unterricht</i>	277
Eva-Sophie Katterfeldt, Nadine Dittert <i>Ein Framework zur Einordnung programmierbarer Baukästen in interdisziplinäre Bildungskontexte</i>	287

Nadine Dittert, Eva-Sophie Katterfeldt <i>Das EduFab-Kit – Ein modularer FabLab-Baukasten für Lehrer*innen . . .</i>	291
Oliver Krisch, Petra Kastl, Ralf Romeike <i>3D-Druck als Motivation für die Einführung in die Programmierung in der Realschule</i>	301
Michael Brinkmeier, Stanislav Pisarenko <i>Nutzung eines Robotiksystems zur Förderung der Berufssprache Deutsch . . .</i>	311
Torsten Brinda, David Tobinski, Stefan Schwinem <i>Schülerinteresse an Informatik und Informatikunterricht</i>	321

Workshops

Katharina Geldreich, Alexandra Funke, Peter Hubwieser <i>Willkommen im Programmierzirkus - Ein Programmierkurs für Grundschulen</i>	327
Otto Thiele, Petra Jückstock <i>Der Hamster hat das Programm erfolgreich beendet – Grundschul Kinder lernen Programmieren</i>	335
Heiko Jochum, Klaus Becker, Martin Zimnol, Daniel Jonietz, Manuel Froitzheim <i>Ein Durchgang durch das Grundfach Informatik (nur) mit dem elektronischen Schulbuch inf-schule.de</i>	345
Barbara Wieczorek, Liz Ribe, Christina B. Class, Michael Brinkmeier <i>Analogien für Programmierkonzepte: Ein Weg zum Computational Thinking</i>	349
Heiko Jochum, Klaus Becker, Martin Zimnol, Manuel Froitzheim <i>Datenbank-Anfragen mit SQL und dem elektronischen Schulbuch inf-schule.de</i>	359
Thomas Rohde, Kolja Strauss, Gregor Alexander Benedikt, René Levens, Lina Haji, Olaf Müller <i>Online kooperieren im Informatikunterricht</i>	363
Esther Alzate Romero, Leonore Dietrich <i>Musikprogrammierung mit Sonic Pi</i>	373

Sven Jatzlau, Ralf Romeike
Herausforderungen durch neue Programmierkonzepte in blockbasierten Programmiersprachen 383

Michael T. Rücker, Nils Pancratz, Carolin Gold-Veerkamp, Niels Pinkwart, Torsten Brinda
Alltagsvorstellungen in der Informatik: Erhebungsmethodik und Implikationen für den Unterricht 393

Poster

Kensuke Akao
Blended-Learning-Module für ein Lehrerfortbildungskonzept zum Informatikunterricht 405

Mike Barkmin, Matthias Kramer, David Tobinski, Torsten Brinda
Unterschiede beim Memorieren von Quelltexten zwischen NovizInnen und ExpertInnen der objektorientierten Programmierung 407

Christine Bescherer, Andreas Fest
Wirkmodelle zum Computational Thinking in der Grundschule 409

Torsten Brinda, Thorsten Terjung
Schülervorstellungen von relationalen Datenbanken: eine empirische Untersuchung 411

Wibke Duwe
Die Handykiste für die Grundschule 413

Christine Günther
Informatik mit und ohne Computer - Praxisideen für Kita, Hort und Grundschule 415

Lennart Goecke, Jurik Stiller, Detlef Pech, Niels Pinkwart
Informatische Grundbildung: Exploration des Erstzugangs zu Lego Wedo 2.0 und Cubelets von Drittklässler_innen 417

Anja Gärtig-Daug, Katharina Weitz, Ute Schmid
Kindliche Modelle der digitalen Welt 419

Mirek Hancl	
<i>Lernwelten verbinden – Coding und Making mit Minecraft</i>	421
Stefanie Jäckel	
<i>Zugänge schaffen und Lernen erleichtern - Motivierung im Informatikunterricht</i>	423
Matthias Kramer, Vahid Samimi, Torsten Brinda	
<i>Entwicklung eines Online-Tools zur Bestimmung objektorientierter Programmierkompetenzen</i>	425
Peter Micheuz, Gerald Futschek	
<i>Der Biber der Informatik in Österreich - Anmerkungen und Perspektiven .</i>	427
Martin Fricke, Kathrin Haselmeier, Ludger Humbert, Dorothee Müller, Daniel Siebrecht	
<i>Informatik im Unterricht – so geht’s</i>	429
Wolfgang Pohl, Robert Czechowski	
<i>Jugendwettbewerb Informatik: BWINF füllt die Lücke</i>	431
Marc Roßner	
<i>Zur Berechnung der Komplexität von einfachen objektorientierten Programmen</i>	433
Arne Saathoff	
<i>Wie stellen sich Schülerinnen und Schüler informatische Konzepte vor und welche sprachlichen Bilder nutzen sie?</i>	435
Daniel Siebrecht	
<i>Sei selbst der Igel! – Umsetzung ausgewählter Elemente der phänomenorientierten Informatik</i>	437
Anke Steinhäuser	
<i>Gezieltes eLearning-Angebot für pädagogische Fachkräfte zur Reduktion von Barrieren bei der Vermittlung von Elementarinformatik im Vor- und Grundschulbereich</i>	439
Maarit Stierle	
<i>Informatik in der Elementarpädagogik – Herausforderung für die Professionalisierung pädagogischer Fachkräfte</i>	441
Manuel Froitzheim, Klaus-Peter Becker, Martin Zimmel, Heiko Jochum	
<i>Förderung digitaler Bildung mit elektronischen Schulbüchern</i>	443

Eingeladene Vorträge

Allgemeinbildung in der digitalen, gestalteten Lebenswelt

Jens Gallenbacher¹

Abstract: Informatik ist allgemeinbildend – das wurde inzwischen deutlich belegt! Dieser Artikel geht der Frage nach, ob in unserer zunehmend gestalteten und durch menschliche Kreativität entstandenen Lebenswelt Schule ohne Informatik überhaupt noch allgemeinbildend sein kann. Auf diesem Wege wird das klassische Peirce'sche Modell der inferentiellen Lerntheorie um die Komponente „Konstruktion“ erweitert.

Keywords: Abduktion, Allgemeinbildung, Allgemeinbildungskriterien, Deduktion, Erkenntnisgewinnungsmethoden, Heymann, Induktion, inferentielle Lerntheorie, Konstruktion, Konstruktivismus, Lebenswelt, Peirce.

1 Einleitung

Vor 14 Jahren zeigte Helmut Witten auf der INFOS 2003 anhand der Aufgaben der allgemeinbildenden Schule nach Heymann auf, dass Informatikunterricht allgemeinbildend ist [Wi2003]. Bereits damals hatte in unserer Lebenswelt ein nie dagewesener Wandel eingesetzt:

Lange Zeit waren die Gesetze der Natur prägender Faktor der Umwelt. Menschen versuchten diese zu verstehen, durch Beobachtung und Schlussfolgerung, aber auch durch Auslegung relevanter Schriften. Wichtige Methoden der Erkenntnisgewinnung waren Induktion, das Ableiten allgemeiner Regeln aus spezifischen Fakten und Deduktion, das Ableiten spezieller Fakten aus allgemeinen Regeln. In der Informatik würden wir von Bottom-up- und Top-down-Verfahren sprechen.

Diese beiden Methoden können alleine allerdings nicht beschreiben, was die immer einflussreicheren Naturwissenschaften hervorbrachten: Eines der wichtigsten geistigen Werkzeuge dieser Disziplinen ist das hypothesengeleitete Experimentieren. Die Hypothese ist keine direkte Schlussfolgerung aus irgendeiner Richtung, daher führte Peirce für die Bildung einer Hypothese den Begriff Abduktion ein [Pe1931]. Das Dreieck nach Abbildung 1 (abgewandelte Form der Version [Mi2015])

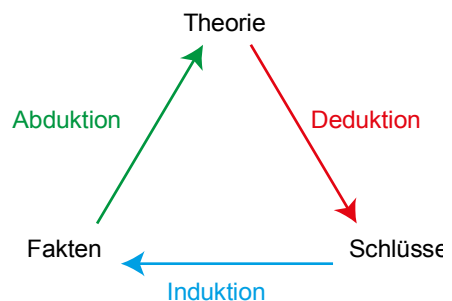


Abbildung 1: Peirce'sches Dreieck

¹ TU Darmstadt, Didaktik der Informatik, Hochschulstraße 10, 64289 Darmstadt, jg@di.tu-darmstadt.de

beschreibt das Modell der inferentiellen Lerntheorie.

Bis etwa Ende des letzten Jahrtausends war diese Abfolge von Abduktion, Induktion und Deduktion auch vollkommen ausreichend für die Erkenntnisgewinnung – zumindest was die Belange der Lebenswelt von Schülerinnen und Schülern angeht: Obwohl immer mehr technische Systeme diese Lebenswelt durchdringen, sind für deren Wirkprinzipien maßgeblich die Gesetze der Natur. So lässt sich etwa für ein sich näherndes Auto trotz „unnatürlichem“ Antrieb mit Hilfe der Kenntnis des Grundprinzips „Kraft = Masse mal Beschleunigung“ abschätzen, ob es sicher ist, die Straße noch davor zu überqueren oder nicht. Selbstverständlich gab es auch bereits im 20. Jahrhundert von Menschen konstruierte Szenarien, für deren Verständnis diese „bekannten“ Gesetzmäßigkeiten nicht sehr hilfreich waren, etwa innerhalb eines Atomkraftwerks. Für die Lebenswelt der Schülerinnen und Schüler spielten diese Ausnahmen allerdings in der Regel keine Rolle, da die Türen zu entsprechenden Bereichen meistens verschlossen waren.



Das hat sich grundlegend verändert mit der zunehmenden Miniaturisierung und Vernetzung elektronischer Komponenten, die die Umwelt zunehmend prägen – erkennbar in Computern, Spielen oder Geräten zur Unterhaltung sowie unsichtbar in Haushaltsgeräten, der Kleidung oder auch „konventioneller“ Technik. Besonders im zwischenmenschlichen Bereich wird immer mehr „virtualisiert“ – Treffen finden oft nicht mehr von Angesicht zu Angesicht statt, sondern mit Hilfe mobil vernetzter Geräte, „Freundschaften“ werden in den sozialen Medien etabliert oder zumindest protokolliert. Aktivitäten „online“ bestimmen zunehmend auch den Erfolg „offline“.

Hierbei handelt es sich demnach um einen sehr relevanten Teil unserer Lebenswelt und Schule hat dies bei der Erfüllung ihrer allgemeinbildenden Aufgaben zu berücksichtigen, wie sie etwa von Heymann sehr gut herausgearbeitet wurden.

Bevor ich im Detail darauf eingehe, möchte ich dies am Beispiel illustrieren.

Diese fiktive Schlagzeile der Abbildung 2 ist keine „Fake News“, sondern beruht durchaus auf einer Studie vom Mai 2011 [Ug2011], in der 721 Millionen Menschen mit 69 Milliarden Freund-Beziehungen untersucht wurden – eine Größenordnung, die für empirische Studien äußerst bemerkenswert erscheint.

Allerdings: Einbezogen wurden alle Teilnehmerinnen und Teilnehmer des Dienstes Facebook, die länger als vier Wochen angemeldet waren und mindestens eine Freund-Beziehung hatten. Die meisten davon wissen übrigens bis heute nichts davon, dass sie an der Studie überhaupt teilgenommen haben. Im Ergebnis wurde tatsächlich festgestellt, dass ca. 84% der „Studienteilnehmerinnen und Studienteilnehmer“ weniger Freunde hatten als der Durchschnitt ihrer Freunde.

Eine wichtige Frage ist allerdings, welche tatsächliche Relevanz diese Erkenntnis hat.

Wir Informatiker kennen für Problemlösung und Problemanalyse das geistige Werkzeug



Abbildung 2: Fiktive Schlagzeile

der Modellbildung und wenn wir dieses für die Fragestellung an einem Beispiel anwenden, kommen wir etwa für eine „normale“ Situation mit 9 Freunden zu einem Graphen wie in Abbildung 3: Knoten repräsentieren die Personen, Kanten die Freundschaftsbeziehungen. Ein sehr kontaktfreudiger Mensch steht hier im Mittelpunkt und hat recht viele Freundschaften. Um ihn herum gibt es mehrere Cliques, die neben ihm nur zwei weitere Freunde kennen.

Damit haben wir eine Person mit acht Freunden und durchschnittlich drei Freundesfreunden sowie acht Personen mit drei Freunden, aber durchschnittlich 4,67 Freundesfreunden. So haben „nachweislich“ acht von neun Personen (also sogar knapp 89%) weniger Freunde als der Durchschnitt ihrer Freunde. Das Beispiel bestätigt also einerseits die Studie, andererseits kann man daran auch relativ leicht erahnen, dass Menschen mit vielen Freunden immer häufiger in die Freundesfreunde-Statistik eingehen als solche mit wenigen Freunden und damit das Facebook-Ergebnis nicht mehr sonderlich

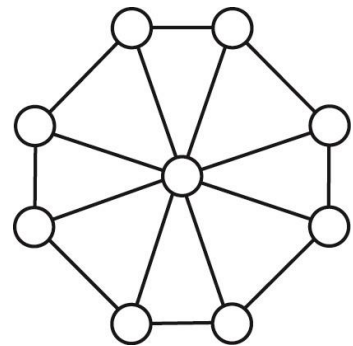


Abbildung 3: Freunde-Graph

überraschend ist.

Zu dem gleichen Ergebnis kommt tatsächlich zwanzig Jahre vor Facebook auch Scott Lauren Feld, der dies als „Freundschaftsparadox“ bezeichnet [Fe1991]: Wenn in einer Gruppe mit Freundschaftsbeziehungen auch nur zwei Personen unterschiedlich viele Freunde haben, hat immer die Mehrheit von Personen weniger Freunde als der Durchschnitt ihrer Freunde.

Dieses Beispiel macht klar, wie wichtig informatisches und ingenieurmäßiges Denken sowie die Kenntnis um informatische Methoden zusammen mit der Kompetenz, diese sinnvoll anzuwenden, sind, um Aussagen richtig interpretieren zu können. Im konkreten Beispiel ist der direkte Einfluss auf die Lebenswelt eher untergeordnet, das Titelbild vom Anfang dieses Artikels sicherlich künstlich. Diese Relevanz ist aber zum Beispiel ganz sicher gegeben im Zusammenhang mit Systemen, die für uns Entscheidungen treffen. Das kann auch die Entscheidung sein, uns mit bestimmten Informationen zu versorgen oder uns diese vorzuenthalten.

So forderte Raj Reddy, Turingpreisträger von 1994, 2016 in Heidelberg in seinem Vortrag „Too Much Information and Too Little Time“ [HLF2016] sogenannte „Guardian Angels“ und „Cognition Amplifiers“ ein. Das sind Softwareprodukte, die für uns die Flut verfügbarer Informationen automatisiert sichten und nur noch interessante Neuigkeiten, Nachrichten und andere Details anzeigen.



Der Kommentar von William Kahan, Turingpreisträger von 1989, auf den Vortrag war „Tell me only, what I want to hear, to reinforce my beliefs, even if they are wrong.“ Er spricht damit an, was wir im deutschen Sprachraum als „Echokammer-Effekt“ oder auch „Filterblase“ kennen. Software auf Mobiltelefonen wie „Siri“ und Geräte wie „Alexa“ dringen einerseits immer weiter in unsere Privatsphäre ein, machen ihre Benutzer transparent und überwachbar. Andererseits gestalten sie zusätzlich aktiv unsere Musikauswahl, unser Medienprogramm, die

Nachrichten, die uns erreichen. Die Entwickler betonen immer wieder, dass dies strikt dem eigenen Profil gemäß erfolge, also nur den – ausgesprochenen und unausgesprochenen – Wünschen der Benutzer entsprechend. Selbst wenn wir in dieser Beziehung den Entwicklern nur beste Absichten unterstellen und Missbrauch ausschließen, kann es aufgrund der Filterblase passieren, dass wir vom System nach einer Zeit nur noch das bekommen, was wir auch hören wollen, ohne uns neuen Aspekten und neuen Ideen öffnen zu müssen. Sehr bequem, aber auch etwas einseitig! Schule muss zumindest die Möglichkeit eröffnen, hier eine informierte Meinung zu bilden.

2 Peirce reloaded

Erkenntnisgewinnung allgemein, insbesondere aber in der Schule, muss sich heute auch mit den Wirkprinzipien gestalteter Systeme auseinandersetzen, denn nur ein Verständnis der Wirkprinzipien erlaubt auch eine mündige Entscheidung – privat für oder gegen die Nutzung bestimmter Medien und Kommunikationskanäle, im Sinne staatsbürgerlichen Handelns aber auch in Bezug auf den möglichen, fürsorglichen regulatorischen Eingriff beim Umgang mit Medien und Systemen.

Wie gewinnt man aber nun unter der Annahme konstruktivistischer Lernpsychologie Erkenntnisse? Das in Abbildung 1 visualisierte Modell inferentieller Lerntheorie nach Peirce beschreibt die Möglichkeiten, naturwissenschaftliche Erkenntnisse zu gewinnen, recht gut:

Abduktion ist die „Kunst“, aus den Beobachtungen der Lebenswelt Hypothesen abzuleiten, die eventuell dazu dienen, das Wissen der Menschheit zu erweitern. Deduktion würden wir Informatiker am ehesten als „top-down“-Verfahren beschreiben: Aus allgemeinen Regeln und Erkenntnissen werden spezielle Details abgeleitet. Induktion wiederum bezeichnet die umgekehrte Vorgehensweise, wenn aus spezifischen Regeln und Beobachtungen allgemeine, weiter greifende Erkenntnisse erschlossen werden. Der Unterschied zur Abduktion ist dabei, dass es sich hier um mehr als eine Theorie bzw. Hypothese handelt.

Diese Methoden der Erkenntnisgewinnung finden wir im Schulunterricht wieder. Auch hier konstruieren Schülerinnen und Schüler im Idealfall ihre Vorstellungen, indem sie aus wahrgenommenen Details allgemeine Regeln ableiten oder indem sie von den allgemeinen Regeln, wie sie etwa aus Schulbüchern recherchiert werden, spezifische Antworten zu aufgeworfenen Fragen ableiten und erweitern so das individuelle Bild ihrer Lebenswelt.

Wenn wir diese beiden Methoden der Erkenntnisgewinnung im Informatikunterricht anwenden, betrachten wir allerdings Computersysteme, Software und die dahinterstehenden Ideen in gleicher Weise wie Naturphänomene: Wie Gegenstände, die entweder unveränderlich vom Universum vorgegeben sind oder zumindest von „Experten“ bzw. „Nerds“, die in der Vorstellung sehr weit weg von der Lebenswelt der Schülerinnen und Schüler sind. Die Betrachtung vieler Unterrichtsmaterialien zu Technik und Informatik zeigt, dass diese Sichtweise von außen im System Schule recht verbreitet ist. Das für die Natur- und Geisteswissenschaften sicherlich meistens sinnvolle Paradigma galt allerdings für Informatik wie für andere Ingenieurdisziplinen im engen Sinne noch nie und es wird immer deutlicher, dass es für die heutige Lebenswelt sogar fehlerhaft ist: Informatiksysteme sind von Menschen gestaltet! Menschen, die altersmäßig oft viel näher an den Schülerinnen und Schülern als an den Lehrpersonen liegen.

Um diese Tatsache im allgemeinbildenden Sinne zu berücksichtigen, lässt sich der verbundene Prozess der Erkenntnisgewinnung mit dem Peirce'schen Modell nicht mehr

sinnvoll beschreiben, denn die Lernenden gehen darüber hinaus, Erkenntnisse zu vorhandenen Fakten zu gewinnen: Vielmehr wird die Lebenswelt um neue Fakten erweitert. Traditionell kennen wir das auch schon aus Unterrichtsfächern wie „Deutsch“ oder „Kunst“, in denen Schülerinnen und Schüler ihre Lebenswelt durch Aufsätze oder Kunstwerke erweitern. Anhand von auf Softwareprodukten beruhenden Systemen wie Facebook, Uber oder auch Rootkits² wird jedoch deutlich, dass die Wirkbreite der Gestaltung durch Informatiksysteme deutlich größer ist.

„Konstruktion“ bildet daher heute neben Abduktion, Deduktion und Induktion eine vierte wichtige Säule der Erkenntnisgewinnung, und könnte das Peirce-Dreieck zum Beispiel in der Art der Abbildung 4 erweitern. Im folgenden Abschnitt wird dies im Licht der Heymann’schen Aufgaben der allgemeinbildenden Schulen betrachtet.

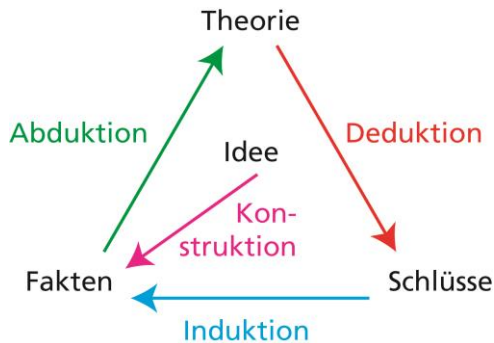


Abbildung 4: Erweitertes Peirce’sches Dreieck

3 Informatik und Allgemeinbildung

Hans Werner Heymann identifiziert in [He2013] fächerunabhängig sieben Aufgaben des allgemeinbildenden Schulsystems: Lebensvorbereitung, Stiftung kultureller Kohärenz, Weltorientierung, Anleitung zum kritischen Vernunftgebrauch, Entfaltung von Verantwortungsbereitschaft, Einübung in Verständigung und Kooperation und Stärkung des Schüler-Ichs. Anhand wichtiger Zitate aus dem Werk wird im Folgenden der Bezug zur Informatik und der gestalteten heutigen Lebenswelt der Schülerinnen und Schüler hergestellt.

Lebensvorbereitung meint demnach die „Vorbereitung auf all das, was Heranwachsende jetzt oder später für ihre Lebensführung in der Gesellschaft, in der sie aufwachsen, mit großer Wahrscheinlichkeit notwendig und unverzichtbar brauchen und

² „root“ bezeichnet bei bestimmten Betriebssystemen Administratorrechte. Das „rootkit“ ist davon abgeleitet ein meistens auch von Laien zu bedienendes Software-Handwerkszeug, das den Einbruch in fremde Systeme ermöglicht. Oft sind in solche rootkits noch Hintertüren eingebaut, die auch den Autoren des Werkzeugs die Möglichkeit einräumen, ein damit kompromittiertes System für eigene Zwecke auszunutzen.

was sie ohne Schule größtenteils nicht lernen würden.“

Betrachten wir Informatiksysteme wie Naturphänomene, können wir sie „verstehen“ lernen und in diesem Zusammenhang sicherlich auch bedienen lernen. Wichtigster Aspekt ist aber das „Begreifen“ der Systeme, was das Nachvollziehen der (von Menschen ausgedachten) Wirkprinzipien impliziert, zusammen mit der Möglichkeit, solche Systeme auch selbst zu gestalten.

Der letzte Punkt ist besonders relevant im Zusammenhang mit der **Stiftung kultureller Kohärenz**: „In allen Fächern sind zentrale Ideen aufzusuchen, mittels derer sich Brücken schlagen lassen zwischen Fach und außerfachlicher Kultur, anhand derer sich deutlich machen lässt, was das Fach (bzw. die korrespondierende Wissenschaft) für die Kulturentwicklung bedeutet, wie es mit ihr verwoben ist, wie es mit dem täglich erfahrbaren gesellschaftlichen Alltag verknüpft ist. [...] In den zentralen Ideen eines Faches [...] verbindet sich der diachrone mit dem synchronen Aspekt der kulturellen Kohärenz: Die zentralen Ideen sind historisch gewachsen, repräsentieren also eine Geschichte; und sie stehen für die Wechselwirkung zwischen Fach und außerfachlicher Kultur, transzendieren also die Grenzen des einzelnen Faches.“

Ebenfalls im direkten Zusammenhang dazu steht die **Einübung in Verständigung und Kooperation**: „Welchen Kurs die hochentwickelten Gesellschaften [...] nehmen können und müssen, darüber dürfen nicht allein Experten befinden, sondern das sind Fragen, die alle angehen und über die alle Staatsbürger - als im Höchstmaß betroffene Laien - mitentscheiden müssen. Die Rationalität derartiger Entscheidungen hängt in hohem Maße von der gelingenden Kommunikation zwischen Laien und Experten ab.“

Viele moderne Wissenschaften – insbesondere Ingenieurwissenschaften und Informatik – arbeiten weitgehend konstruktiv gestaltend. Ein Unterrichtsfach, das diesen Aspekt der Erkenntnisgewinnung aufgreift, ist daher obligatorisch! Andernfalls besteht die Gefahr zweier Parallelkulturen: Auf der einen Seite die der „Nerds“, die (aufgrund außerschulischer Bemühungen) Einblick in die Gestaltung moderner Systeme haben und als „Technokraten“ de facto entscheidende Teile der Lebenswelt gestalten und bestimmen. Auf der anderen Seite die Kultur der „Normalen“, zu denen die gewählten Vertreter der Mehrheit der Bürgerinnen und Bürger gehören, die aber keine aktiven und auch keine passiven Kompetenzen bei der Gestaltung der Systeme haben, sondern lediglich Anwender sind.

Heymann: „Für die Idee der **Weltorientierung** ist ein gewisses ‚Luxurieren‘ des Wissens charakteristisch, gestützt auf die Annahme, dass ein differenziertes Weltbild einen weiten Urteilshorizont erschließt und sowohl die Klärung des eigenen Standortes in der Welt wie auch seine Relativierung erleichtert.“

Dieses „Luxurieren“ zeichnet sich in der heutigen Lebenswelt durch immer kürzere Innovationszyklen aus, die das System Schule zunehmend überfordern. Statt mit Hilfe traditioneller Erklärmodellen den Innovationszyklen hinterherzuhinken, sollte Weltorientierung durch Erweiterung dieser Erklärmodelle um die Konstruktion und die

gestalterischen Aspekte ermöglicht werden und auf diese Weise der Charakter der Innovation selbst als relevanter Lerngegenstand ins Curriculum einziehen.

Zum **kritischen Vernunftgebrauch** schreibt Heymann: „Wer seine Vernunft kritisch gebrauchen kann, wird weder ein bequemer Untertan sein, der fraglos hinnimmt, was Lehrer, Politiker und andere Vertreter institutioneller Macht in der öffentlichen Verwaltung, in Kirche, Wirtschaft und Wissenschaft ihm zumuten, noch wird er sich umstandslos einem vermeintlichen oder faktischen Druck der „öffentlichen Meinung“ beugen. Kritische Vernunft birgt politische Sprengkraft in sich - insbesondere angesichts „unvernünftiger“ Verhältnisse.“

Die im letzten Abschnitt erwähnte „Echokammer“ legt nahe, dass heute ergänzend zu Verwaltung, Kirche, Wirtschaft und Wissenschaft noch informatisch bzw. ingenieurmäßig gestaltete Systeme betrachtet werden müssen, die – gegebenenfalls ganz ohne Hintergedanken ihrer Entwickler – Einfluss auf Meinungsbildung nehmen oder aber auch in Zeiten postfaktischer Politik und Fake-News ganz bewusst zur Manipulation ausgenutzt werden können. Kritischen Vernunftgebrauch rein durch die Betrachtung von außen, aus der Bediener Sicht zu erreichen, wird erfahrungsgemäß sehr schwierig! Viel effizienter ist, wenn eine gewisse Durchdringung der Gestaltungsprinzipien erreicht wurde und Basis für die kritische Reflexionskompetenz der Schülerinnen und Schüler ist.

Die **Entfaltung von Verantwortungsbereitschaft** grenzt Heymann folgendermaßen ab: „Verantwortliches Handeln setzt beides voraus, Gefühl und Wissen. Das Verantwortungsgefühl allein kann mich nicht hindern, im Sinne der Verantwortungsethik, die mir die Folgen meines Handelns zurechnet, unverantwortlich zu handeln, wenn ich nämlich zu wenig weiß. Das unverantwortliche Handeln ist dann uninformiertes oder dummes Handeln. Wissen und kritischer Vernunftgebrauch allein können mich andererseits nicht hindern, gewissenlos zu handeln, wenn mir Verantwortungsgefühl und Verantwortungsbewusstsein abgehen. Das unverantwortliche Handeln ist in diesem Falle böses, zynisches oder zumindest leichtfertiges Handeln.“

Schule sollte hier mindestens dafür sorgen, dass sich niemand auf die erste Position zurückziehen kann, indem Probleme, die vermeintlich aus dem gestalteten Teil der Lebenswelt entstehen der alleinigen Verantwortung der „Nerds“ und damit „der anderen“ zugeordnet werden! Auch hierfür ist daher im Unterricht ein Minimum an Einblick in die Gestaltung selbst nötig.

Eine der wichtigsten Aufgaben allgemeinbildender Schulen ist sicher die **Stärkung des Schüler-Ichs**: „Ich-Stärkung zielt auf die Entwicklung von Selbstbewusstsein, Selbstvertrauen, personaler Identität, auf die Fähigkeit, eigene Ziele, Wünsche und Vorstellungen klar zu erkennen und handelnd zu verwirklichen, mit den eigenen Stärken und auch Schwächen realistisch umzugehen.“

Die Möglichkeit, ohne großen finanziellen Aufwand Systeme gestalten zu können, die unsere Lebenswelt und die Art zu leben sowie als Gesellschaft zusammenzuleben

nachhaltig verändern ist – unabhängig davon, ob sich die Schülerinnen und Schüler dafür entscheiden, diese Möglichkeit zu nutzen – wahrscheinlich die bedeutendste Komponente für die Stärkung des Schüler-Ichs. Eine allgemeinbildende Schule muss dafür Sorge tragen, dass diese Möglichkeit nicht nur Kindern aus sogenannten „bildungsnahen“ Schichten zur Verfügung steht, die sich das Prinzip der Konstruktion und Gestaltung auch außerhalb der Schule aneignen können. Alle sollen von diesen Möglichkeiten profitieren und selbstbewusst erwachsen werden, mit dem Vertrauen auf das eigene Potential zur Gestaltung.

4 Resümee

Helmut Witten kam bereits 2003 berechtigterweise zum Ergebnis, dass Informatik allgemeinbildend ist. Für die heutige Lebenswelt und die mit ihrer stetigen Veränderung verbundene immer weiter steigende Dominanz gestalteter Systeme muss erweiternd festgestellt werden:

Eine allgemeinbildende Schule ist ohne die Anerkennung und Berücksichtigung des wachsenden Anteils der von Menschen (mit-)gestalteten Lebenswelt nicht möglich! Im konstruktivistischen Lernprozess ist somit die Erkenntnisgewinnungsmethode „Konstruktion“ als wichtiges allgemeinbildendes Element einzubeziehen.

Für die Vermittlung der abgeleiteten Kompetenzen sind grundständig in informatisch- bzw. ingenieurmäßig-gestalterischem Denken ausgebildete Lehrpersonen obligatorisch, was zwingend ein eigenständiges Unterrichtsfach voraussetzt. Informatik erfüllt diese Bedingungen als einziges bundesweit. Die Forderung nach einem bundesländerübergreifenden Pflichtfach Informatik ist daher gleichzeitig die Forderung nach einer modernen allgemeinbildenden Schule.

Allgemeinbildende Schule ist heute ohne das für alle Schülerinnen und Schüler verbindliche Pflichtfach Informatik mit angemessenem Stundenumfang nicht möglich!

5 Literaturverzeichnis

- [Fe1992] Scott L. Feld: Why do your friends have more friends than you do?, American Journal of Sociology, 96(6), 1991
- [He2013] Hans Werner Heymann: Allgemeinbildung und Mathematik. Studien zur Schulpädagogik und Didaktik, Bd. 13. Weinheim/Basel: Beltz 1996 - Neuauflage 2013
- [HLF2016] Raj Reddy: Too Much Information and Too Little Time, Vortrag auf dem Heidelberg Laureate Forum am 22. September 2016, online unter http://www.heidelberg-laureate-forum.org/event_2016/

- [Mi2015] Gerhard Minnameier: Tightening the Peirce-Strings Forms of Abduction in the Context of an Inferential Taxonomy in Magnani, L. & Bertolotti, T. (Eds.) (2015), Springer handbook of modelbased science. Berlin: Springer
- [Pe1931] Charles Sanders Peirce: The Collected Papers of Charles Sanders Peirce Vols. I-VI ed. Charles Hartshorne and Paul Weiss (Cambridge, MA: Harvard University Press, 1931-1935), Vols. VII-VIII ed. Arthur W. Burks (same publisher, 1958) in der Electronic Edition 1994
- [Ug2011] Johan Ugander, Brian Karrer, Lars Backstrom, Cameron Marlow: The Anatomy of the Facebook Social Graph, eprint arXiv:1111.4503, 11/2011
- [Wi2003] Helmut Witten: Allgemeinbildender Informatikunterricht? Ein neuer Blick auf H. W. Heymanns Aufgaben allgemeinbildender Schulen in Informatische Fachkonzepte im Unterricht, INFOS 2003, 10. GI-Fachtagung Informatik und Schule, 17.-19. September 2003 in Garching bei München, Seiten 59-75

Unterrichtsqualität in der digitalen Welt

Hilbert Meyer¹

Ich bin von Haus aus Bildungstheoretiker und kein Mediendidaktiker. Deshalb bewege ich mich bei diesem Vortrag auf dünnem Eis. Das von der Konferenzleitung vorgeschlagene Thema habe ich zu folgender Frage konkretisiert: Welche Ansprüche an die Gestaltung des Informatikunterrichts und an die Arbeit mit digitalen Medien in anderen Fächern folgen aus dem aktuellen Diskussionsstand zur Unterrichtsqualität? Der Vortrag besteht aus vier Abschnitten:

In der Einleitung (Abschnitt 1) wird kurz skizziert, welche Diskursverweigerungen (Hartmut von Hentig), welche schlecht fundierten Attacks (Manfred Spitzer) und welche behutsamen Annäherungsversuche (Klaus Zierer) Didaktiker und neurowissenschaftliche Hobbydidaktiker im Blick auf digitale Bildung und digitalen Medieneinsatz im Unterricht in den letzten Jahren vorgenommen haben. Ich plädiere dabei für größtmögliche Nüchternheit. Die Digitalisierung der Welt ist weit fortgeschritten. Es geht nicht mehr um die Frage „Wollen wir sie reinlassen?“, sondern um die Frage: Wie können digitale Medien besser als bisher für selbstreguliertes Lernen und für transformationale Bildungsprozesse (Hans-Christoph Koller) genutzt werden?

Im Abschnitt 2 („Was ist guter Unterricht?“) wird anhand von vier empirisch basierten Kriterienkatalogen zum „guten Unterricht“ (von Meyer, von Helmke, von der Forschergruppe COACTIV, von Good et al.) analysiert, dass sich diese Kataloge weltweit annähern. Mein Fazit: Ein übergeordnetes Kriterium ist und bleibt für mich die Frage, ob im Unterricht sinnstiftendes Kommunizieren stattgefunden hat.

Im Abschnitt 3 („Was sagt die Forschung?“) erläutere ich beispielhaft an einigen wenigen Forschungsergebnissen, worauf bei Urteilen über Unterrichtsqualität zu achten ist:

1. Wir müssen zwischen Oberflächen- und Tiefenstrukturen unterscheiden.
2. Wir sollten zur Kenntnis nehmen, dass sich der Unterrichtsprozess, wie schon von Johann F. Herbart postuliert, in einem fortwährenden Wechsel von Vertiefung und Besinnung entfalten.
3. Deshalb hat der von der Lehrperson angeleitete Metaunterricht auch im Unterricht mit und über digitale Medien eine zentrale Bedeutung.

¹ Hilbert Meyer, Prof. em., Kastanienallee 40, 26121 Oldenburg

4. Wir sollten dabei aber beachten, dass Lehrerinnen und Lehrer Weltmeister im Kompensieren von Schwächen im einen Kompetenzfeld durch Stärken im andern sind.

Im Abschnitt 4 geht es um die Frage: „Was tun?“ Ich votiere gegen ein eigenes Unterrichtsfach „Informatik“ in der Grundschule, aber für ein solches Fach in den Sekundarstufen. Ich skizziere mein Drei-Säulen-Modell der Unterrichtsentwicklung (aus Meyer 2015) und plädiere dafür, die drei Grundformen der Direkten Instruktion, des Individualisierenden und des Kooperativen Unterrichts zu unterscheiden. In allen drei Grundformen können digitale Medien eine wichtige Rolle spielen – aber immer kommt es darauf an, die Lehrer-Schüler-Interaktionen zu stärken. Dieser Abschnitt endet mit dem Motto: „Mischwald ist besser als Monokultur.“

Ein Handout mit der schriftlichen Fassung des Vortrags wird zu Beginn des Vortrags verteilt.

Keywords: Unterrichtsqualität; Informatikunterricht; Medien im Unterricht; sinnstiftendes Kommunizieren

Verstehen – Vernetzen – Verantworten

Warum *Medienbildung* und *informatische Bildung* uns alle angehen und wir sie gemeinsam weiterentwickeln sollten

Thomas Knaus¹

Abstract: *Digital* ist ein Begriff, der uns heute in sämtlichen Lebensbereichen begegnet. Das „Digitale“ ist aber nicht nur ein omnipräsenter *Hashtag*, sondern bezeichnet technologische und technische Entwicklungen, die in hohem Maße subjekt- und gesellschaftsprägend sind. Diese sozialisatorische Relevanz des Digitalen soll im vorliegenden Beitrag anhand von fünf Thesen exemplarisch aufgezeigt werden: So beeinflussen digitale Medien unsere Wahrnehmung und damit die *produktive Verarbeitung* von Wirklichkeit. Aufgrund der ständigen Verfügbarkeit (mobiler) digitaler Medien, Werkzeuge und technischer Netze umgeben uns auch unsere *sozialen* Netzwerke permanent, was neue Formen sozialer Eingebundenheit sowie interessen- und kontextbezogene Vergemeinschaftungen ermöglicht. Da wir mit digitalen Medien nicht nur Umwelt wahrnehmen, sondern zunehmend auch *selbst* Realität in Form von digitalen Artefakten und Werkstoffen *erzeugen*, verschmelzen einerseits unsere „analogen“ und „virtuellen“ Realitäten; andererseits werden Realitäts*konstruktionen* unmittelbar erlebbar. Hiervon kann die Entwicklung von Subjekt und Gesellschaft nicht unbeeindruckt bleiben – denn digitale Medien und Werkzeuge sind inzwischen in sämtlichen Sozialisationsinstanzen allgegenwärtig und entwickeln sich aufgrund ihrer neuen sozialen Bedeutung zunehmend vom *Interface* zum *kommunizierenden Gegenüber*. Auslöser von subjekt- und gesellschaftskonstituierenden Konstruktionsprozessen sind also nicht nur andere Menschen, sondern auch deren mediale Artefakte – *Dinge*, die mittels digitaler Werkzeuge *produziert* und *manipuliert* werden und Individuen in zunehmendem Maße kommunikativ vernetzen. Digitale Medien sollten daher sowohl als *Mittel* als auch als *Gegenstand* einen zentralen Platz im schulischen Unterricht einnehmen. Zudem zielt die Medienbildung auf die Bildung des Subjekts. Die gesellschaftliche Teilhabe setzt künftig das mediengebildete Subjekt voraus. Doch wird es nicht mehr *nur* in der Hand der oder des Einzelnen liegen, Kontrolle über die Werkstoffe der Zukunft, die Daten und Algorithmen, zu behalten. Da die Prozesse hinter dem *Interface* der Maschine für Menschen nicht mehr unmittelbar durchschaubar und kontrollierbar sind, ist gleichermaßen auch eine Sensibilisierung für Technik und technische Prozesse desiderat, die hinter den digitalen Medien und in den Werkzeugen stecken. Denn unsere sozialen Realitäten, ob digital oder analog, sind aktiver und individueller denn je *gestaltbar* – in dieser Gestaltbarkeit steckt gleichermaßen ein zu nutzendes Potential sowie eine zu erbringende Pflicht. Aus diesen Potentialen und sich eröffnenden Gestaltungsspielräumen heraus erwächst unsere Verantwortung, Medienerziehung und informatische Bildung idealerweise *gemeinsam* weiterzudenken.

Keywords: Medienpädagogik, Bildungsinformatik, digitale Medien, Medienbildung, informatische Bildung, „digitale Bildung“, „digitales Lernen“, Schule, Mensch-Maschine-Interaktion, Kommunikation, digitale Werkzeuge und Werkstoffe, Software, Programmierbarkeit, Adaptivität, Vernetzung, Konvergenz, Symbolischer Interaktionismus, Sozialisation, Sozialisationstheorien, Mediensozialisation, KMK-Strategie, Dagstuhl-Dreieck.

¹ Wissenschaftlicher Direktor des FTzM der Frankfurt University of Applied Sciences, Nibelungenplatz 1, 60318 Frankfurt am Main, knaus@ftzm.de.

1 Turn: Digital, medial – egal?

Digital ist das neue *Bio* – zumindest scheint es so, als habe das Adjektiv „digital“ vieles ersetzt, was kürzlich noch als „medial“ bezeichnet wurde. So wird aktuell vielerorts diskutiert, ob in einer *digital vernetzten Welt* [GI16] auch „digitale Bildung“ benötigt wird [Ba17]. Sollte daraufhin auch unsere Disziplin in *Digitalpädagogik* umbenannt werden? Unabhängig von der noch zu klärenden Frage, ob Bildung überhaupt „digital“ sein kann, sollten wir uns das „omnipräsente Adjektiv“ [Kn16b], das derzeit so unkritisch in nahezu alle gesellschaftlichen, kulturellen, ökonomischen und politischen Vollzüge übernommen wird, etwas genauer anschauen. Denn – so viel sei an dieser Stelle bereits verraten – die begriffliche Unterscheidung zwischen *Medium* und digitalem *Werkzeug* erleichtert das *Verständnis* der sozialen Funktion und der sozialisatorischen Bedeutung digitaler Technik.

Die Kritik am aktuellen *Hashtag* und am *omnipräsentem Adjektiv* soll vorliegend mittels einer alternativen fachlichen Sichtweise befeuert werden: Wenn Informatikerinnen und Informatiker das Adjektiv *digital* hören, denken sie zunächst an Einsen und Nullen. In einer binären Welt lässt sich aus einem Schaltzustand heraus *alles* konstruieren. Irgendwie mag diese Komplexitätsreduktion jedoch nicht zu dem passen, was wir mit „digitalen Welten“, „digitaler Gesellschaft“ oder auch „digitaler Bildung“ ausdrücken wollen. *Digitalisierung* bezeichnet üblicherweise die Übertragung analoger Erscheinungsformen in diskrete Werte mit dem Ziel, diese elektronisch zu verarbeiten, zu distribuieren oder zu speichern. Eine „analoge Gesellschaft“ lässt sich entsprechend nicht einfach in eine digitale Gesellschaft *transformieren*. Ist also der Begriff der Digitalisierung zutreffend, wenn er auf eine Gesellschaft angewendet wird – auf Kultur, auf Bildungsfragen, auf Schule oder auf das menschliche Lernen?

Eigentlich müssten die Fragen doch lauten: Wie verändern sich unser Leben, Zusammenleben und unsere Umwelt in einer technologisch und technisch induzierten gesellschaftlichen Umbruchsituation und welche Fähigkeiten werden für das Aufwachsen, die Integration und ein souveränes Miteinander benötigt? „Wie wirken digitale Medien auf Individuen und die Gesellschaft, wie kann man Informationen beurteilen, eigene Standpunkte entwickeln und Einfluss auf gesellschaftliche und technologische Entwicklungen nehmen? Wie können Gesellschaft und Individuen digitale Kultur und Kultivierung mitgestalten?“ [GI16]. Zugegeben, *Hashtags* wie „digitales *Something*“ sprechen sich schneller und prägen sich leichter ein. Doch trifft beispielsweise „digitale Bildung“ wirklich den Kern dessen, was wir damit meinen? Eine präzisere Analyse erscheint notwendig. Im Folgenden sollen fünf Thesen die Relevanz digitaler Medien für die Sozialisation in *digital vernetzten* Gesellschaften aufzeigen.

2 Sign: Thesen zur sozialisatorischen Relevanz des Digitalen

Die folgenden fünf Thesen sollen die *Zeichen der Zeit* markieren und bilden die Basis für die Re-Interpretation eines klassischen Konzepts (vgl. Kapitel 3), das unter Berücksichtigung derzeitiger technologischer und technischer Entwicklungen, die in hohem Maße subjekt- und gesellschaftsprägend sind, aktueller denn je erscheinen wird.

2.1 Digitale Medien beeinflussen Umweltwahrnehmung (1. These)

Nikki Yanofsky sagte vor Kurzem im *ZEITmagazin*, dass sie als Zwölfjährige aus lauter Neugier „Jazz“ googelte und daraufhin eine bekannte Jazz-Sängerin wurde. Hieran erstaunt eigentlich gar nichts. Dieses Beispiel – das auch ein ganz anderes sein könnte – soll lediglich zeigen, welche Relevanz digitale Medien für unsere *Weltwahrnehmung* haben. Dass die Relevanz „virtueller“ Wahrnehmung mitunter gewichtiger sein kann als *face-to-face*-Begegnungen, zeigt das Beispiel ebenfalls, berücksichtigt man nämlich, dass Yanofsky in *Montreal* aufgewachsen ist – also just in *der* Stadt, in der jährlich das weltweit größte Jazz-Festival stattfindet. Um mit Jazz in Berührung zu kommen, hätte Nikki alternativ auch vor die Tür gehen können. Es kann konstatiert werden: Digitale Medien beeinflussen in hohem Maße unsere *Weltwahrnehmung* und entsprechend die „produktive Verarbeitung von Wirklichkeit“ [Hu06].

2.2 Digitale Werkzeuge sind omnipräsent (2. These)

Studien wie die KIM- und JIM-Studien des MPFS [MP14; MP16] zeigen, dass fast alle Menschen inzwischen über einen persönlichen Zugriff auf digitale Geräte verfügen. So sind laut der JIM-Studie 2016 Jugendliche in Deutschland mit Handy, Smartphone, Computer und Internetzugang nahezu *voll* ausgestattet [MP16]. Diese umfassende Verfügbarkeit digitaler Geräte wird vielerorts als Initial eines tiefgreifenden gesellschaftlichen Wandels beschrieben [Aß16; Ba07; Ca14; DI14; Ho17; KM16; Sc14]. Doch was änderte sich mit der rasanten Verbreitung des *mobilen* und *vernetzten* Computers – in Gestalt des Smartphones – wirklich? Ist ein *technisches* Gerät in der Lage die *soziale* Welt zu verändern?

Der historische Technikeinsatz war im Wesentlichen durch die Erleichterung *physischer* Arbeit gekennzeichnet: So revolutionierte beispielsweise die Eisenbahn die Mobilität; im Bausektor machte der Technikeinsatz das kräftezehrende Handwerk nicht nur sicherer und für Menschen erträglicher, sondern ermöglichte auch völlig neue Bauwerke. Technik hatte aber auch direkten Einfluss auf das soziale Miteinander: „Die Einführung der Sprache konstituierte die Stammesgesellschaft, die Einführung der Schrift die antike Hochkultur [und] die Einführung des Buchdrucks die moderne Gesellschaft [...]“ [Ba07]. Technik revolutionierte also die physische Arbeit und Medien, wie die Schrift und der Buchdruck sowie der Computer, reformieren die geistige Arbeit [Na92; Sc97] und sie

prägten als jeweils dominierende Kommunikationsmedien das gesellschaftliche Miteinander.

Es geht außerdem nicht um *ein* Gerät, denn die Mensch-Computer-Relation veränderte sich seit der Einführung des Computers entscheidend: So arbeiteten noch vor relativ kurzer Zeit *mehrere* Personen an *einem* Computer – in Banken arbeiteten Menschen an Terminals; in Universitäten konnten Forschende Rechenkapazitäten des Großrechners buchen. In den letzten drei Jahrzehnten dominierte der „persönliche“ (Personal) Computer – der PC – und damit die *Eins-zu-eins*-Relation von Mensch und Maschine. Heute genügt vielen von uns nicht mehr nur ein Gerät. Damit änderte sich das (zunächst rein quantitative) Verhältnis von Mensch und Maschine erneut. Zur steigenden Zahl digitaler Geräte, die einen Menschen umgeben, gesellen sich zu den sichtbaren Geräten, wie Smartphone, Tablet und Notebook, auch weniger auffällige *Computer in Alltagsgegenständen*, wie Motorsteuerungen und Bremsassistenten im Auto, Haus- und Heizungssteuerungen oder auch „smarte“ Kühlschränke und Zahnbürsten. In nicht allzu ferner Zukunft wird beinahe jeder Gegenstand des täglichen Lebens einen Computer enthalten und mit den anderen Gegenständen vernetzt sein, was als IoT – *Internet of Things* – bezeichnet wird und die *Omnipräsenz des Digitalen* zementiert.

Es geht aber nicht nur um Quantitäten, sondern um eine neue *Qualität* von Technik und damit auch um ein neues Verständnis der Mensch-Maschine-*Interaktion* – nachfolgend weitere Thesen dazu.

2.3 Digitale Werkzeuge ermöglichen neue Formen sozialer Vernetzung (3. These)

Digitale – und vor allem *mobile* – Medien ermöglichen eine soziale Eingebundenheit in neuer Quantität und Qualität [Kn16c; Kn17]. Noch vor wenigen Jahren war der *ständige* Zugang zum Internet und seinen Diensten unüblich; dann wurde das erste Smartphone mit einer *Internet-Flatrate* für Mobilfunknetze angeboten und vieles änderte sich: Dank der technischen Vernetzung konnte nicht nur die verhältnismäßig leistungsschwache Hardware des mobilen Geräts mittels online bereitgestellter Speicher- und Prozessor-Ressourcen ergänzt werden, womit die mobile Technik performanter wurde. Es wurden darüber hinaus auch Menschen miteinander „vernetzt“ [Wi02]. Mit dieser Vernetzung sind keineswegs nur technische Plattformen im WWW (Soziale Netzwerke) gemeint, sondern die Tatsache, dass das Individuum permanent mit seinen *sozialen* Netzwerken und *Peers* – wie der Familie und dem Freundeskreis – verbunden sein kann und sich aufgrund dieser dauerhaft verfügbaren *kommunikativen Vernetzung* neue persönliche Netzwerke entwickeln können [Kr16]. War noch vor wenigen Jahren die soziale Vernetzung in hohem Maße durch *räumliche Nähe* geprägt, so entscheiden künftig vor allem Themen oder individuelle *Interessen* – beispielsweise in Form von *Communities of Practice* [We98] – über persönliche Verbundenheit und soziale Eingebundenheit. Durch die Konvergenz von physischen und virtuellen Wirklichkeiten können wir also Grenzen überwinden und damit neue Formen der Vergesellschaftung erleben.

2.4 Digitale Werkzeuge *machen* Realität (4. These)

An der Benutzerschnittstelle, am *Interface* – also an der Stelle, an der Technik zum *Gegenüber* wird – ist es zunächst irrelevant, ob es sich um ein analoges oder digitales Medium handelt [Je06], denn die mittels analoger Technik hergestellten Bilder, Filme oder Tondokumente stehen bezüglich ihrer medialen Erscheinung digital produzierten Dokumenten in nichts nach [Kn09]. Während das Medium primär die äußere Erscheinungsform, als *Objekt der Wahrnehmung* [Ke06] die Schnittstelle zum Menschen darstellt, umfasst der Werkzeugbegriff auch die zugrunde- und dahinterliegende technische Basis des digitalen Geräts, die aufgrund ihrer *Programmierbarkeit*, *Adaptivität*, *Konvergenz* und technischen *Vernetzung* umfassende Gestaltungsmöglichkeiten der medialen Erscheinungsformen sowie der eigenen Funktionalitäten bietet und damit über Potentiale verfügt, die über jene der analogen Medien hinausreichen [Kn15a; Kn16a/b; Kn17].

Weiterführender als die Differenzierung analoger und digitaler Technik ist also vielmehr die Unterscheidung der Technik in Bezug auf ihre *soziale Funktion*: Erscheint sie primär als *Medium* oder als *Werkzeug*? Gerade bezüglich *digitaler* Technik mag diese Unterscheidung auf den ersten Blick künstlich oder gar antiquiert wirken, zumal sie zunehmend verschwimmt und Menschen doch von Anbeginn ihre dingliche Umwelt mittels Werkzeugen gestalteten. Die Unterscheidung zwischen Medium und Werkzeug ist aber gerade unter Berücksichtigung der zuvor benannten *Spezifika des Digitalen* wesentlich und weiterführend, da sie den Blick auf die soziale Bedeutung und Funktion der Technik schärft: So sind beispielsweise Werkzeuge im Gegensatz zu Medien ohne aktiv Nutzende „nutzlos“ [Kn15b]. Die Werkzeug-Analogie verdeutlicht darüber hinaus, dass das Medium nicht nur dann beteiligt ist, wenn wir Umwelt *wahrnehmen*, wie zuvor beispielhaft mit der Jazzsängerin Nikki Yanofsky erläutert (vgl. Kapitel 2.1), sondern auch dann, wenn wir unsere Umwelt sowie digitale Werkstoffe mittels digitaler Werkzeuge *gestalten*.

Die Produktion und aktive Gestaltung *analoger* (Massen-)Medien war zwar grundsätzlich möglich, aber stets aufwändig – man denke nur an die historische Vielfältigkeit von Texten oder die ersten Filme aus den Anfängen der Kinematografie. Erst mit dem Aufkommen der Digitaltechnik konnten auch diejenigen, die üblicherweise rezipieren, das Objekt der Wahrnehmung *direkt* beeinflussen – es *manipulieren*, wie es der Informatiker Reinhard Keil nennt: „Zum ersten Mal in unserer Mediengeschichte kann das Objekt der Wahrnehmung auch unmittelbar zum Objekt der Manipulation werden“ [Ke06]. Damit kann das digitale Medium nicht nur rezipiert werden, sondern auch ohne größere Aufwände selbst *produziert* und „interaktiv“ *gestaltet* werden. Hierbei ist wesentlich, dass der alltagssprachlich eher negativ besetzte Begriff der *Manipulation* nicht falsch verstanden wird: Es ist hier nicht der aus dem Französischen stammende Fachbegriff aus der Soziologie oder Psychologie gemeint – es geht also *nicht* um gezielte oder verdeckte Einflussnahme oder um Konditionierung. Und es geht auch nicht darum, dass Medien Menschen „manipulieren“. Der Begriff der Manipulation sollte vielmehr in seinem *eigentlichen Wortsinn* verstanden werden: So ist

manus das lateinische Wort für „Hand“ und *plere* bedeutet „füllen“. Man könnte also wörtlich übersetzen: „eine Handvoll haben“ oder etwas passender: „die Dinge in der Hand haben“. Damit ist gemeint, dass wir *digitale* Medien nicht nur wahrnehmen können, sondern auf das Objekt der Wahrnehmung auch *selbst handelnd zugreifen* können.

2.5 IT doesn't matter ... Software takes command (5. These)

Der Werkzeugbegriff erscheint als Relikt einer vergangenen Zeit – der Hardwarezeit. Dabei verfügt die *physische* IT, obwohl sie *die* technische Basis alles Digitalen ist, heute nur noch über eine nachrangige Bedeutung. Um es mit dem bekannten Zitat von Lev Manovich „Software takes command“ [Ma08] auszudrücken: Die *Software hat die Macht* übernommen. Anhand mobiler Computer, lässt sich die schwindende Relevanz von Hardware und die steigende Bedeutung von Software und technischen Netzen gut veranschaulichen: Smartphones wären nichts ohne die *Apps*, die wir individuell zusammenstellen und mittels der die Hardware zu einem ganz persönlichen Werkzeug – einem „digitalen Taschmesser“ – vervollkommen wird. Auch die Strategien der globalen Internetkonzerne wie *Amazon*, *Apple* und *Google* lassen diese Bedeutungsverschiebung erahnen: Der Hardware wird in Relation zum *Content*, wie Apps, Musik, Videos etc., auch ökonomisch eine immer geringere Bedeutung beigemessen. Die Relevanz des Netzes lässt sich anhand von Apps zur Sprachsteuerung (wie beispielsweise *SIRI*) illustrieren: Smartphones sind nur deswegen so leistungsfähig, weil die rechenintensiven Prozesse, wie Sprachanalyse, inhaltliche Auswertung und Antwortgenerierung, nicht innerhalb des mobilen Gerätes stattfinden, sondern die Speicher- und Prozessorleistung verteilter Server im Netz genutzt werden. Auch dies ist gemeint, wenn in Anlehnung an Lev Manovich konstatiert wird, dass die Software und – ich ergänze – das Netz² „die Macht übernehmen“ [Ma08].

Digitale Medien ermöglichen also aufgrund ihrer Programmierbarkeit und technischen Vernetzung die Integration von Werkzeugen zur aktiven Gestaltung und Verknüpfung von Inhalten innerhalb einer Einheit – häufig innerhalb eines Geräts oder einer Plattform. Diese digitalen Werkzeuge tragen demnach nicht nur dazu bei, unterschiedliche codale und mediale Elemente zu verbinden, sondern darüber hinaus auch Lerngegenstand und das (didaktische) Werkzeug selbst miteinander zu verschmelzen [Kn15b]. Damit können nicht nur digitale Medien selbst produziert und

² Bestätigt wird diese Bedeutungsverschiebung durch aktuelle Studien, wie die bereits zitierten KIM- und JIM-Studien des MPFS [MP16] und die Studie zur *Generation What*, einer europaweiten Jugendstudie von ZDF, SWR und dem Bayerischen Rundfunk. Für letztere Studie wurden junge Erwachsene nach ihren Einstellungen und unter anderem auch zur *Relevanz des Netzes* in ihrem Leben befragt – mit interessantem Ergebnis: So können sich 80 Prozent der befragten 18- bis 34-jährigen Menschen in Deutschland ein Leben ohne Gott, 70 Prozent ein Leben ohne Auto und 79 Prozent ein Leben ohne Fernsehen vorstellen. Selbst ein glückliches Leben ohne Kinder ist immerhin für 52 Prozent der Befragten denkbar – aber: „nicht ohne Internet“: Laut <http://www.generation-what.de/portrait/data/all> (aufgerufen am 21. Mai 2017) können sich lediglich 48 Prozent der Befragten in Deutschland ein glückliches Leben ohne Internet vorstellen, unter den 16- und 17-Jährigen sind es nur 42 Prozent.

manipuliert werden, sondern digitale Werkzeuge verfügen auch über eine hohe funktionale und formale *Adaptivität* – vor allem aufgrund theoretisch unendlicher Erweiterungs- und Kombinationsmöglichkeiten mittels individuell zusammenstellbarer Apps und gegebenenfalls auch entsprechender Peripherie. Dies bedeutet, dass nicht nur Inhalte beziehungsweise (Lern-) Gegenstände gestaltbar werden, sondern auch das (Lern-) Werkzeug selbst.

3 Re-Turn: Digitale Werkzeuge und ihre Bedeutung für subjekt- und gesellschaftskonstituierende Interaktion

Zusammenfassend kann festgehalten werden, dass digitale Medien in hohem Maße unsere *Wahrnehmung von Umwelt* und damit die *produktive Verarbeitung von Wirklichkeit* beeinflussen (vgl. Kapitel 2.1). Aufgrund der *Omnipräsenz* digitaler Medien (vgl. Kapitel 2.2) und der ständigen Verfügbarkeit technischer Netze umgeben uns auch unsere *sozialen* Netze permanent (vgl. Kapitel 2.3), was neue Formen sozialer Eingebundenheit und eine interessen- und kontextbezogene Vergemeinschaftung ermöglicht. Da wir zunehmend auch mittels *digitaler* Medien Realität erzeugen, kann *Realitätskonstruktion* in unmittelbarer Weise *erlebt* werden und es können in der Revision digitaler Artefakte Reflexionsprozesse angestoßen werden. Damit sind digitale Medien und soziale Netze nicht nur ständig beteiligt, wenn wir andere und uns selbst wahrnehmen und inszenieren, sondern aufgrund der grundsätzlichen Speicherbarkeit digitaler Medien stehen mediale Konstruktionen zur Selbstreflexion zur Verfügung (vgl. Kapitel 2.4). Prozesse von *Sozialisation* werden demnach zunehmend *medial vermittelt* und *medial konstruiert*: Die Wahrnehmung von innen und außen, die Konstruktion von Umwelt und die Subjektbildung sind von Medien, die wohlgernekt omnipräsent sind, durchsetzt und beeinflusst [Su99; Ca14; Kn17]. Wenn man dies voraussetzt, dann verwundert es, dass in den Sozialisierungstheorien digitale Medien noch immer eine so untergeordnete Rolle spielen und bisher in der Sozialisierungsforschung bestenfalls als *Massenmedien* wahrgenommen werden [Hu06; Hu08]. In den kommunikationswissenschaftlichen und medienpädagogischen Arbeiten zur „Mediensozialisation“ [Fr03; Ho10; Vo10] werden – zu diesem Schluss kommt Dieter Spanhel – „die Medien in ihrer Bedeutung für den Sozialisierungprozess isoliert oder unter eingeschränkten Perspektiven konzeptualisiert“ [Sp13]. So werden Medien mitunter als *Orte der Sozialisation* bezeichnet und damit nur sehr spezifische Funktionen der Medien für die Persönlichkeitsentwicklung betont, obwohl inzwischen Einigkeit darüber herrscht, dass sich das Verständnis von Medien als Sozialisierungsinstanz – also *neben* Familie und Schule – überholt hat, da Medien inzwischen in *allen* Instanzen vorhanden und prägend sind (vgl. Kapitel 2.2).

Nach den für die deutschsprachige Sozialisierungsforschung prägenden Autoren Dieter Geulen und Klaus Hurrelmann entsteht und entwickelt sich die Persönlichkeit eines Menschen „in wechselseitiger Abhängigkeit von der gesellschaftlich vermittelten sozialen und materiellen Umwelt“ [Ge80]. Damit wird als *Sozialisation* die Summe aller

Lernprozesse bezeichnet, die der Mensch in seiner Entwicklung zu einer handlungsfähigen Persönlichkeit durchläuft. Wenn nun also, wie zuvor konstatiert, das Leben zwar „analog“, Kommunikation und Kollaboration aber zunehmend „digital“ ist und entsprechend *medial* stattfindet [Kn16c], dann kann Persönlichkeitsentwicklung hiervon nicht unbeeindruckt bleiben [Ca14].

Nach dem *Symbolischen Interaktionismus* [Bl69; Me73], einer Theorie, die für die handlungs- und rollentheoretische Perspektive als wesentliche soziologische Basistheorien innerhalb des Sozialisationskonzepts prägend war [Hu06], ist Kommunikation zwischen Individuen der Schlüssel zur Subjektkonstitution; Kommunikation dient demnach nicht bloß dem Transport von Botschaften, sondern primär der *Konstitution* und *Entwicklung* von Subjekt und Gesellschaft [Sc14]. Charles Horton Cooley, einer der geistigen Väter von George Herbert Mead, nimmt an, dass die Impulse für die Persönlichkeitsentwicklung *vom Anderen* ausgehen. Diese Erkenntnis wurde möglicherweise durch Johann Wolfgang von Goethe inspiriert: „Der Mensch erkennt sich nur im Menschen“ [Go99]. Für Cooley findet die Entwicklung von Persönlichkeit mittels *Kommunikation* statt, er schreibt dazu: „It is through communication that we get our higher development. The faces and conversation of our associates; books, letters, travel, arts [...] supply the stimulus and framework for all our growth“ [Co72]. Mead entwickelte in Anlehnung an Cooley ein Modell, das die Konstitution des Subjekts durch Kommunikation konzeptionalisiert; von seinem Schüler Herbert Blumer wurde dieses Modell dann weiter ausgebaut. Darin entsteht die Persönlichkeit, von Mead und Blumer als *Self* bezeichnet, als Produkt zweier Größen: Dem „sozialen Selbst“, dem *Me*, und der psychischen Komponente des *I*.

Das *Me* steht für die in einer Gesellschaft existierenden gemeinsamen Normen, Regeln, Werthaltungen, also: für die sozialen Konsense [Me73]. Es präsentiert die Vorstellung dessen, wie die anderen Menschen – also: das soziale Umfeld – ein Individuum sehen und wie es sich nach der Interpretation ihrer Erwartungen zu verhalten habe. Das *Me* wiederum „speichert“ die intersubjektiv ausgehandelten Erwartungen und stellt darüber hinaus handlungsleitende Strukturen und Orientierungen zur Verfügung. Das *I* vertritt – gegenüber dem *Me* – impulsive und spontane Energien der Person, die zwar durch das *Me* „gezügelt“ werden, aber doch eine unabhängige Größe der Persönlichkeit darstellen. Durch ebendieses Zusammenwirken von *I* und *Me* bildet sich das *Self*, also: das *Selbst* – das Selbstverständnis. Dieses Zusammenspiel, so der Kommunikationswissenschaftler Friedrich Krotz, müsse man sich als *kommunikativen Prozess* vorstellen [Kr98]. Ein Prozess, der nach außen hin wahrnehmbar sein könne, sich aber durchaus auch als „innerer Dialog“ abspiele.

Cooley, Mead und Blumer sind sich darin einig, dass diese „sozialisierende“ Kommunikation nicht nur zwischenmenschlich erfolgt, sondern auch *zwischen Mensch und Dingen* möglich ist. Kommunikation findet also auch dann statt, wenn *Dinge* uns ein Handeln auferlegen oder nahelegen. Ein Beispiel hierfür ist der *Berliner Schlüssel* [La93]: Dieser Schlüssel mit zwei identischen Bärten wird nach dem Aufschließen des Schlosses durch das Schloss hindurchgeschoben und ermöglicht daraufhin das

Zuschließen der Durchgangstür von der anderen Seite. Dieser Schlüssel ist ein sehr treffendes Beispiel für Dinge, die Menschen ein bestimmtes Handeln auferlegen – man könnte auch sagen: *Dinge, die „kommunizieren“* [Kn09]. Auch digitale Medien, Werkstoffe und Werkzeuge kommunizieren und können so unser Handeln beeinflussen. In Bezug auf die Massenmedien hat der kanadische Philosoph Marshall McLuhan mit der vielzitierten Erkenntnis *the Medium is the Message* darauf hingewiesen, dass Medien nicht neutral und demnach niemals nur „Behältnisse“ für Informationen sind [Mc68]. Dass wir heute im Gegensatz zum Buch, Film, Fernsehen oder der Architektur *direkten* und *kreativen* Einfluss auf das *digitale* Medium selbst nehmen können, in dem wir es aktiv *gestalten*, konnte er seinerzeit bestenfalls ahnen.

4 Connect: Das Gestaltende als Verbindung zwischen Informatik und Pädagogik

Bei der Entscheidung für mein Studium folgte ich meinem persönlichen Interesse. Dabei verband ich – wie damals noch recht unüblich – eine Ingenieurwissenschaft mit der Sozial- und Medienpädagogik (und ergänzte später noch Erziehungswissenschaft und Pädagogische Psychologie). Dass sich die Medienpädagogik auf die geistes- und sozialwissenschaftliche Erziehungswissenschaft und die Informatik als Struktur- und Ingenieurwissenschaft auf eine andere Logik der Theoriebildung beruft [Br17], empfand ich mitunter als irritierend und inspirierend zugleich. Die beiden Professionen erscheinen also auf den ersten Blick kaum gegensätzlicher – sie verfügen jedoch über mindestens eine verbindende Gemeinsamkeit: die *Gestaltung*.

Während die Informatik als die Wissenschaft der systematischen Organisation, Speicherung und elektronischen Verarbeitung von Informationen gilt und sich beispielsweise innerhalb der Technischen Informatik mit der Gestaltung von Algorithmen und Architekturen befasst, gestalten Pädagoginnen und Pädagogen unter anderem erzieherische Interventionen und (Lern-) Umfelder. Wenn – im Sinne des bereits in alternativer Übersetzung zitierten Buchtitels von Lev Manovich – die *Software* es ist, die *Befehle annimmt* [Ma08], dann könnten diese Aufgabentrennungen tendenziell verschwimmen, da potentiell alle Menschen künftig ohne größere Aufwände (wie es beispielsweise noch in einer hardwaredominierten Zeit oder vor der Einführung grafischer Benutzeroberflächen war) *Technik gestalten* können [Sc2011; Kn15b]. Am Beispiel von Content-Management-Systemen lässt sich beispielhaft darlegen, dass zur Gestaltung einer ansprechenden Webseite primär gute Ideen, geeignete Texte und gelungene Grafiken sowie Bilder benötigt werden, jedoch keine HTML- und Programmierkenntnisse mehr. Auch digitale Werkzeuge, wie das Smartphone, lassen sich ohne tiefergehendes technisches Know-how individualisieren. Software wird damit zum *Werkstoff*, aus dem sich hilfreiche *Werkzeuge* herstellen, weiterentwickeln und kombinieren lassen. Programmierbare und adaptive digitale Werkzeuge sowie verbesserte Benutzerschnittstellen und gemeinsame Codes erleichtern damit einerseits die interdisziplinäre Zusammenarbeit, andererseits ermöglichen sie *Making* – das kreative

Gestalten mittels digitaler Werkzeuge und -stoffe: Wie wäre es wohl, wenn beispielsweise Lehrende nicht die (didaktischen) Medien und Werkzeuge nutzen müssten, die ihnen angeboten werden, sondern sie diese *selbst* gestalten könnten?

5 De-Sign: Verantworten? Gestalten!

Wir sollten uns ein Beispiel an digitaler Technik nehmen und – wie vorliegend beschrieben – auch selbst Grenzen überwinden, uns neu vernetzen und dadurch Tradiertes *re-formieren*.

Routinierte Abläufe und Gewohnheiten sind für uns Menschen zwar wesentlich, denn trainierte Handlungsweisen und -abläufe benötigen keine weitere Aufmerksamkeit mehr und ermöglichen die Konzentration auf die wesentlichen Dinge: Nahmen das *Schleifpunktsuchen* sowie „Spiegel, Blinker, Schulter“ früher unsere ganze Konzentration in Anspruch, beherrschen wir diese Handlungen heute ganz nebenbei. Nicht selten jedoch stehen just jene tradierten Handlungspraxen einem Weiterkommen im Weg, wie der fünfte *fraMediale*-Band am Beispiel des *digitalen Wandels in Bildungseinrichtungen* zeigt [Kn16d]. Zur Illustration ein kleines Experiment: Bitte falten Sie Ihre Hände. Prüfen Sie bitte, ob bei Ihnen der rechte oder der linke Daumen oben liegt. Nehmen Sie nun die Hände wieder auseinander. Bitte falten Sie nun *noch einmal* Ihre Hände, aber nun absichtlich so, dass Ihr *anderer* Daumen oben liegt. Was Sie nun spüren, ist die *Macht der Gewohnheit* und damit erhalten wir einen vagen Eindruck davon, warum digitale Medien so schwer Eingang in Schule und Hochschule finden. Was ist zu tun?

5.1 Schule erneuern: Kooperation und Making

Die Medienpädagogik formuliert nicht nur gut begründete Anforderungen zur Notwendigkeit eines Lernens *über* Medien an Schulen [Kn16b], sondern kann gleichermaßen auch aktivierende und kreative Zugänge für das Lernen *mit* Medien, die Förderung von Medienbildung und informatischer Bildung anbieten. Hilfreich hierfür wäre die Infragestellung der traditionellen Grenze zwischen der Institution Schule und der außerschulischen Medien (projekt-)arbeit – die bisher kaum zu überwindende Demarkationslinie zwischen formalen, non-formalen und informellen Bildungskontexten. Durch medienpädagogische Projektarbeit und Aktive Medienarbeit nach subjekt- und handlungsorientierten Ansätzen, wie sie u. a. Bernd Schorb [Sc95], Dieter Baacke [Ba96] und Gerd Tulodziecki [Tu97] propagieren, wird die Chance eröffnet, digitale Medien als Verbindungsglied für etwas zu nutzen, was nicht unbedingt voneinander getrennt sein muss: die Lebenswelt der Schülerinnen und Schüler und das (schulische) Lernen. Hiermit würde nicht nur die Rolle der Schule als *Lebensraum* befördert, sondern den Schülerinnen und Schülern könnten neue anknüpfbare Erfahrungsräume bereitgestellt werden [Ni04; Wa16]. Wie ich in anderen Arbeiten umfassender untersuchte und ausführte [Kn15a; Kn16c; Kn17], können beispielsweise

aktivierende und lebensweltverbindende Potentiale digitaler Medien schon heute in der Schule genutzt werden; technische und gesellschaftliche Entwicklungen ermöglichen einerseits, erfordern andererseits aber mitunter auch, bestehende organisatorische Grenzen wie tradierte Stundentaktungen, Fächergrenzen und Raumkonzepte infrage zu stellen und neu zu denken.

5.2 Gemeinsam Gesellschaft gestalten: Medienbildung ist nicht genug

Medienbildung zielt auf die Bildung des Subjekts. Sie soll umfassend sein, das mediengebildete Subjekt ist und bleibt in einer *digital vernetzen Welt* [GI16] relevant. Doch zunehmend liegt es nicht mehr *nur* in der Hand der oder des Einzelnen, Kontrolle über die eigenen (personenbezogenen) Daten zu behalten. Eine Person kann sich noch so überlegt im WWW und in Sozialen Netzwerken bewegen: Wenn unscheinbare *Apps* persönliche Daten sammeln – wer liest schon die langen Nutzungsvereinbarungen (die sich mit jedem Update ändern können) – ist es mit der „Datensouveränität“ passé. Auch wenn ein Nutzer im Adressbuch eines anderen *WhatsApp*-Kontakts gespeichert ist, kann das jeweilige Individuum nicht mehr *souverän* und fallbezogen entscheiden, ob die Speicherung und Weitergabe persönlicher Daten erfolgen darf – respektive einschätzen, ob eine solche Entscheidung überhaupt möglich ist. *Ergänzend* zur subjektiven Medienbildung sollte ein *gesellschaftlicher Diskurs* darüber stattfinden, was Technik und global operierende beziehungsweise vernetzte Unternehmen dürfen. Der Diskurs muss in Regeln und (Selbst-) Verpflichtungen – oder allgemein: Normen – möglichst transparent und nachvollziehbar kodifiziert werden. Da das Netz und darin operierende Unternehmen Staatsgrenzen überwinden, sollten auch diese Normen *grenzübergreifend* gedacht, diskutiert und vereinbart werden.

Medienkompetenz ist ein Begriff, der in seiner traditionellen Auffassung recht wenig mit Technik zu tun hatte [Sc07] und der interessanterweise heute im Alltagsverständnis nicht selten sogar ausschließlich auf technische Fertigkeiten *verkürzt* wird [Kn16b]. Dabei ging es doch Dieter Baacke, der den Begriff in Anlehnung an das Kompetenzkonzept von Noam Chomsky und die Arbeiten von Karl-Otto Apel und Jürgen Habermas prägte, nicht um die Technik, die Geräte oder die Medien, sondern um *Kommunikation und Kooperation* [Ba73; Ba96]. Die Fähigkeit eines Menschen also, Medien, (digitale) Werkzeuge und Technik im Allgemeinen souverän für eigene Ziele und Bedürfnisse zu nutzen und selbst-, medien- und gesellschaftsbezogen zu reflektieren, kreativ und partizipativ zu handeln sowie analytisch und strukturell Wissen zu erwerben [Ba96]. So geht es auch heute nicht um Technik, sondern um die Frage, wie und wozu wir digitale Werkzeuge, Werkstoffe und Medien nutzen (wollen) und damit im Wesentlichen noch immer um *Kommunikation* und *Kooperation* – nur eben in ihrer digitalen Form. Da nun gerade *digitale* Technik mehr ist als ein bloßer „Mittler“, sondern sie auch an der *Herstellung* von (Medien-) Inhalten [Sc07] und Kommunikation [Ba07] beteiligt ist und selbst Interpretationen vornimmt [Kn16b] – wie semantische Technologien und beispielhaft die „Antwortmaschine“ *Wolfram|Alpha* –, rückt ein weiteres Ziel von Medienkompetenz in den Fokus der Betrachtung: die Kenntnis der technischen und

organisatorischen Bedingungen der Wissens- und Medienproduktion.

Wichtig ist also, dass idealerweise *alle* Menschen in der Lage sind, die Technik selbst, die Algorithmen, auf deren Grundlage sie arbeitet, sowie generell Prozesse der Entstehung und Reproduktion von Wissen zumindest grundlegend nachzuvollziehen [Kn16b]. Nur wer über die Kenntnis verfügt, was hinter der Benutzerschnittstelle – hinter dem *Interface* der Maschine – vor sich geht, kann kompetent rezipieren und souverän kommunizieren. Daher gehören zu einer umfassenden Medienkompetenz ein konzeptionelles Technikverständnis und mindestens grundlegende Aspekte informatischer Bildung [GI16; KM16; Kn16b; Sc16]. Im Sinne der zuvor geforderten Grenzüberwindungen sollten daher Medienbildung und informatische Bildung in geeigneter Form [Re15; He16; Sc16; Tu16b] miteinander verbunden werden. Exemplarisch möchte ich im Folgenden auf zwei aktuelle Entwicklungen hinweisen, die Medienbildung und informatische Bildung „gemeinsam denken“.

5.3 Gemeinsam weiterdenken: *Dagstuhl-Erklärung* zum Verhältnis von informatischer Bildung und Medienbildung

Die *Dagstuhl-Erklärung* 2016 wurde von Informatikerinnen und Informatikern sowie Medienpädagoginnen und Medienpädagogen *gemeinsam* verfasst [GI16]. Von beiden Seiten wurde die Redaktion des Dokuments als herausfordernd beschrieben und auch nach seiner Veröffentlichung wird es weiter diskutiert und kritisiert – das *muss* so sein, denn dies ist wesentliches Kennzeichen von Wissenschaft. Die Definition eines gemeinsamen Gegenstands – auch wenn dieser als *Hashtag* in den Communities nicht nur Zustimmung findet (vgl. Kapitel 1) – und das Aufeinanderzubewegen fanden einen überfälligen und guten Anfang. Im Juli 2017 wird der gemeinsame Austausch zwischen Informatik und Medienpädagogik im Rahmen einer Klausurtagung zum „Verhältnis von Medienbildung und informatischer Bildung: Dagstuhl-Dreieck *in progress*“ [KB17] im FTzM in Frankfurt am Main fortgesetzt.

5.4 Gemeinsam Potentiale des Digitalen nutzen: KMK-Strategie zur „Bildung in der digitalen Welt“

Ein weiteres aktuelles Beispiel für das gemeinsame Weiterdenken von informatischer Bildung und Medienbildung findet sich in der Strategie der Kultusministerkonferenz (KMK) zur „Bildung in der digitalen Welt“ [KM16], die am 8. Dezember 2016 in Berlin beschlossen und deren Entwurf seitens der Fachgruppe Schule der GMK³, der Initiative KBoM!⁴ und der Sektion Medienpädagogik der DGfE⁵ kritisch kommentiert wurde [GM16; KB16]. Unsere wesentlichen Kritikpunkte waren die Verwendung eines

³ GMK: Gesellschaft für Medienpädagogik und Kommunikationskultur, <http://www.gmk-net.de>

⁴ KBoM!: Initiative Keine Bildung ohne Medien, <http://www.keine-bildung-ohne-medien.de/>; siehe auch Veröffentlichungen u. a. von Horst Niesyto zum Wirken und den Zielen der Initiative [Ni16]

⁵ DGfE: Deutsche Gesellschaft für Erziehungswissenschaft, <http://www.dgfe.de>

unklaren und funktionalistischen Bildungsbegriffs sowie die schwerlich nachvollziehbare Tatsache, dass die jüngste Erklärung hinter den Zielen des ausgefüllten Papiers zu „Medienbildung in der Schule“ aus dem Jahr 2012 zurückblieb. Seitens des GI-Fachbereichs *Informatik und Ausbildung/Didaktik der Informatik* (IAD) wurde kritisiert, dass „eine Beschränkung auf die Nutzung digitaler Medien in allen Unterrichtsfächern bedeuten [würde], die digitale Welt so zu akzeptieren, wie sie jetzt ist, und deren Weiterentwicklung anderen zu überlassen. Schule muss aber zu gesellschaftlicher Teilhabe befähigen“ [Br16].

Was an der neuen KMK-Erklärung als sehr positiv bewertet werden kann, ist die Tatsache, dass *erstmal*s ein Papier der KMK zur *Vorab*-Kommentierung veröffentlicht wurde. Das heißt, die Erklärung behandelt nicht nur im gegenständlichen Sinne die Herausforderungen, die digitale Technik an die Gesellschaft und das Bildungssystem stellt, sondern die Kultusministerkonferenz nutzte mit dem vorherigen Einbezug der Fachgesellschaften auch das *Potential*, die digitale Technik zur Organisation kollaborativer und partizipativer Prozesse bietet. Dies erscheint mir in Anbetracht vieler weiterer komplexer Aufgaben, die eine *fachübergreifende* Expertise und *transnationale* Beteiligung erfordern, als ein kluger und weiterhin verfolgungswürdiger Weg.

Literaturverzeichnis

- [Ar65] Arnheim, Rudolf; Bock, Henning: Kunst und Sehen. Eine Psychologie des schöpferischen Auges (Art and visual perception). De Gruyter, Berlin, 1965.
- [AB16] Aßmann, Sandra; Brüggem, Niels; Dander, Valentin; Gapski, Harald; Sieben, Gerda; Tillmann, Angela; Zorn, Isabel: Digitale Datenerhebung und -verwertung als Herausforderung für Medienbildung und Gesellschaft. In (Brüggemann, Marion; Knaus, Thomas; Meister, Dorothee, Hrsg.): Kommunikationskulturen in digitalen Welten. kopaed, München, S. 131-139, 2016.
- [Ba07] Baecker, Dirk: Studien zur nächsten Gesellschaft. Suhrkamp, Frankfurt am Main, 2007.
- [Ba17] Bastian, Jasmin; Feldhoff, Tobias; Harring, Marius: Call for Papers zum Themenheft „,Digitale Bildung‘. Medienbezogene Bildungskonzepte für die ‚nächste Gesellschaft‘“, http://medienpaed.com/public/journals/1/cfps/CfP_Digitale_Bildung.pdf, 02.05.17
- [Ba73] Baacke, Dieter: Kommunikation und Kompetenz. Grundlegung einer Didaktik der Kommunikation und ihrer Medien. Juventa, München, 1973.
- [Ba96] Baacke, Dieter: Medienpädagogik. Grundlagen der Medienkommunikation (Bd. 1). Niemeyer, Tübingen, 1996.

- [Be02] Benjamin, Walter: Medienästhetische Schriften (darin: Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit). Suhrkamp, Frankfurt am Main, 351-383, 2002.
- [Bl69] Blumer, Herbert: Symbolic interactionism: Perspective and method. Prentice-Hall, New Jersey (USA), 1969.
- [Br16] Brinda, Torsten: GI kritisiert reduzierten Bildungsbegriff der KMK: Informatik muss in Bildungsstrategie für die digitale Welt integriert werden (25. Juli 2016), <https://www.gi.de/aktuelles/meldungen/detailansicht/article/gi-kritisiert-reduzierten-bildungsbegriff-der-kmk-informatik-muss-in-bildungsstrategie-fuer-die-dig.html>, 02.05.2017.
- [Br17] Brinda, Torsten; Diethelm, Ira; Kommer, Sven; Rummler, Klaus: Call for Papers zum Themenheft „Medienpädagogik und Didaktik der Informatik. Eine Momentaufnahme disziplinärer Bezüge und schulpraktischer Entwicklungen“, http://medienpaed.com/public/journals/1/cfps/CfP_Medienpaedagogik_und_Infomatik.pdf, 02.05.17.
- [Ca14] Carstensen, Tanja; Schachtner, Christina; Schelhowe, Heidi; Beer, Raphael: Subjektkonstruktionen im Kontext digitaler Medien. In (Carstensen, Tanja; Schachtner, Christina; Schelhowe, Heidi; Beer, Raphael, Hrsg.): Digitale Subjekte, Transcript, Bielefeld, 2014.
- [Co72] Cooley, Charles Horton: Social Organization. A Study Of The Larger Mind. Schocken Books, New York, 1972.
- [Ge80] Geulen, Dieter; Hurrelmann, Klaus: Zur Programmatik einer umfassenden Sozialisationstheorie. In (Hurrelmann, Klaus; Ulich, Dieter, Hrsg.): Handbuch Sozialisationsforschung, Beltz, Weinheim, 51-67, 1980.
- [GI16] GI – Gesellschaft für Informatik (2016), Bildung in der digitalen vernetzten Welt (Dagstuhl-Erklärung): <https://www.gi.de/aktuelles/meldungen/detailansicht/article/dagstuhl-erklaerung-bildung-in-der-digitalen-vernetzten-welt.html>, 14.07.2016.
- [GM16] GMK – Gesellschaft für Medienpädagogik und Kommunikationskultur: Stellungnahme der FG Schule der GMK zum Strategiepapier der Kultusministerkonferenz vom 12. Mai 2016 zu „Bildung in der digitalen Welt“, http://www.gmk-net.de/fileadmin/pdf/GMK-Stellungnahme_zum_KMK-Strategie-Entwurf.pdf, 02.05.2017.
- [Go99] Goethe, Johann Wolfgang von: Poetische Werke (Bd. 5), Phaidon Verlag, Essen, 1999.

- [He16] Herzig, Bardo: Medienbildung und Informatische Bildung – Interdisziplinäre Spurensuche. *MedienPädagogik* 25/16, 59-79, 2016, <http://www.medienpaed.com/article/view/428>, 02.05.2017.
- [Ho10] Hoffmann, Dagmar; Mikos, Lothar: *Mediensozialisationstheorien – Modelle und Ansätze in der Diskussion*. VS, Wiesbaden, 2010.
- [Hu06] Hurrelmann, Klaus: *Einführung in die Sozialisationstheorie*. Beltz, Weinheim, 2006.
- [Hu08] Hurrelmann, Klaus; Grundmann, Matthias; Walper, Sabine: *Handbuch Sozialisationsforschung*. Beltz, Weinheim, 2008.
- [Je06] Jenkins, Henry: *Convergence Culture. Where Old and New Media Collide*. New York University Press, New York, 2006.
- [KB16] KBoM – Keine Bildung ohne Medien (2016): Stellungnahme der Initiative Keine Bildung ohne Medien – KBoM! zum Strategiepapier der Kultusministerkonferenz vom 12. Mai 2016 zu „Bildung in der digitalen Welt“, <http://www.keine-bildung-ohne-medien.de/publications/stellungnahme-zum-kmk-strategiepapier-bildung-in-der-digitalen-welt>, 02.05.2017.
- [KB17] KBoM – Keine Bildung ohne Medien (2017): Klausurtagung zum Verhältnis von informatischer Bildung und Medienbildung im Juli 2017 – Dagstuhl-Dreieck in progress (Tagungsankündigung), <http://ftzm.de/medienbildung/klausurtag-zum-verhaeltnis-von-informatischer-bildung-und-medienbildung>, 21.05.2017.
- [Ke06] Keil, Reinhard: Zur Rolle interaktiver Medien in der Bildung. In (Keil, Reinhard; Schubert, Detlef, Hrsg.): *Lernstätten im Wandel – Innovation und Alltag in der Bildung*. Waxmann, Münster, 59-77, 2006.
- [Ke17] Keiner, Edwin: Didaktik – Bildung – Technik – Kritik. *Medienpädagogik und Antinomien der Moderne*. *MedienPädagogik* 27/17, 270-286, 2017, <http://www.medienpaed.com/issue/view/33>, 02.05.2017.
- [KM16] KMK – Kultusministerkonferenz (2016): *Bildung in der digitalen Welt*, https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2016/Bildung_digitale_Welt_Webversion.pdf, 10.12.2016.
- [Kn09] Knaus, Thomas: *Kommunigrafie*. kopaed, München, 2009.
- [Kn15a] Knaus, Thomas: Me, my Tablet – and Us. Vom Mythos eines Motivationsgenerators zum vernetzten Lernwerkzeug für autonomopoietisches Lernen. In (Friedrich, Katja; Siller, Friederike; Treber, Albert, Hrsg.): *Smart und mobil – Digitale Kommunikation als Herausforderung für Bildung, Pädagogik und Politik*. kopaed, München, 17-42, 2015.

- [Kn15b] Knaus, Thomas; Engel, Olga: (Auch) auf das Werkzeug kommt es an – Technikhistorische und techniktheoretische Annäherungen an den Werkzeugbegriff in der Medienpädagogik. In (Knaus, Thomas; Engel, Olga, Hrsg.): fraMediale – digitale Medien in Bildungseinrichtungen (Bd. 4). kopaed, München, 15-57, 2015.
- [Kn16a] Knaus, Thomas: Potentiale des Digitalen – Theoretisch-konzeptionelle Betrachtungen pädagogischer und didaktischer Potentiale des schulischen Einsatzes von Tablets und BYOD. medien+erziehung: schule. smart. mobil 60/16, 33-39, 2016.
- [Kn16b] Knaus, Thomas: digital – medial – egal? – Ein fiktives Streitgespräch um digitale Bildung und omniprésente Adjektive in der aktuellen Bildungsdebatte. In (Brüggemann, Marion; Knaus, Thomas; Meister, Dorothee, Hrsg.): Kommunikationskulturen in digitalen Welten. kopaed, München, 99-130, 2016.
- [Kn16c] Knaus, Thomas: Kooperatives Lernen. Begründungen – digitale Potentiale – konzeptionelle Perspektiven. In (Scheer, August Wilhelm; Wachter, Christian Hrsg.): Perspektiven für die digitale Weiterbildung – Bildungslandschaften der Zukunft. IMC AG, Saarbrücken, 141-155, 2016.
- [Kn16d] Knaus, Thomas; Engel, Olga: Wi(e)derstände – Digitaler Wandel in Bildungseinrichtungen (Bd. 5 der fraMediale-Reihe). kopaed, München, 2016.
- [Kn17] Knaus, Thomas: Pädagogik des Digitalen – Phänomene – Potentiale – Perspektiven. In (Eder, Sabine; Micat, Claudia; Tillmann, Angela, Hrsg.): Software takes command, kopaed, München, 2017.
- [Kr98] Krotz, Friedrich: Digitalisierte Medienkommunikation: Veränderungen interpersonaler und öffentlicher Kommunikation. In (Neverla, Irene, Hrsg.): Das Netz-Medium. Kommunikationswissenschaftliche Aspekte eines Mediums in Entwicklung, Westdeutscher Verlag, Opladen, 113-136, 1998.
- [Kr16] Krotz, Friedrich: Wandel von sozialen Beziehungen, Kommunikationskultur und Medienpädagogik – Thesen aus der Perspektive des Mediatisierungsansatzes. In (Brüggemann, Marion; Knaus, Thomas; Meister, Dorothee M., Hrsg.): Kommunikationskulturen in digitalen Welten. kopaed, München, 19-42, 2016.
- [La93] Latour, Bruno: Der Berliner Schlüssel. Erkundungen eines Liebhabers der Wissenschaften (La clef de Berlin et autres leçons d'un amateur de sciences). Akademie-Verlag, Berlin, 1996.
- [Ma01] Manovich, Lev: The Language of New Media: MIT Press, Cambridge, 2001.
- [Ma08] Manovich, Lev: Software takes Command, www.softwarestudies.com/softbook, 02.05.2017.

- [Mc68] McLuhan, Marshall: Die Gutenberg Galaxis: Das Ende des Buchzeitalters. Econ, Düsseldorf, 1968.
- [Me73] Mead, George Herbert: Geist, Identität und Gesellschaft. Suhrkamp, Frankfurt am Main, 1973.
- [MP14] MPFS – Medienpädagogischer Forschungsverbund Südwest: KIM 2014 – Kinder + Medien, Computer + Internet. MPFS, Stuttgart, 2014.
- [MP16] MPFS – Medienpädagogischer Forschungsverbund Südwest: JIM 2016 – Jugend, Information, (Multi-) Media: MPFS, Stuttgart, 2016.
- [Na92] Nake, Frieder: Informatik und die Maschinisierung von Kopfarbeit. In (Coy, Wolfgang; Nake, Frieder; Pflüger, Jörg-Martin; Rolf, Arno; Seetzen, Jürgen; Siefkes, Dirk; Stransfeld, Reinhard, Hrsg.): Sichtweisen der Informatik. Vieweg, Braunschweig, 181-201, 1992.
- [Ni04] Niesyto, Horst: Öffnung von Schule und partnerschaftliche Kooperation. Zur Zusammenarbeit von schulischer und außerschulischer Medienarbeit. In (Pöttinger, Ida; Schill, Wolfgang; Thiele, Günter, Hrsg.): Medienbildung im Doppelpack. kopaed, München, 39-49, 2004.
- [Ni16] Niesyto, Horst: Keine Bildung ohne Medien! – Kritische Medienbildung jenseits funktioneller Vereinnahmung. In (Knaus, Thomas; Engel, Olga, Hrsg.): Wi(e)derstände – Digitaler Wandel in Bildungseinrichtungen (Bd. 5 der fraMediale-Reihe). kopaed, München, 17-34, 2016.
- [Re15] Reiter, Anton (2015): Medienbildung und informatische Bildung, <http://medienimpulse.at/articles/view/795>, 02.05.2017.
- [Sc07] Schelhowe, Heidi: Technologie, Imagination und Lernen – Grundlagen für Bildungsprozesse mit Digitalen Medien. Waxmann, Münster, 2007.
- [Sc14] Schachtner, Christina; Duller, Nicole: Kommunikationsort Internet. Digitale Praktiken und Subjektwerdung. In (Carstensen, Tanja; Schachtner, Christina; Schelhowe, Heidi; Beer, Raphael, Hrsg.): Digitale Subjekte. Transcript, Bielefeld, 81-154, 2014.
- [Sc16] Schelhowe, Heidi (2016): Through the Interface – Medienbildung in der digitalisierten Kultur. MedienPädagogik 25/16, 41-58, 2016, <http://www.medienpaed.com/article/view/427/426>, 02.05.2017.
- [Sc95] Schorb, Bernd: Medienalltag und Handeln. Medienpädagogik in Geschichte, Forschung und Praxis. Leske+Budrich, Opladen, 1995.
- [Sc97] Schelhowe, Heidi: Das Medium aus der Maschine. Zur Metamorphose des Computers. Campus, Frankfurt am Main, 1997.

- [Sp13] Spanhel, Dieter (2013): Sozialisation in mediatisierten Lebenswelten – Grundzüge eines theoretischen Bezugsrahmens. *merz-Wissenschaft* 57.6/13, 30-43, 2013.
- [Su99] Sutter, Tilmann: Systeme und Subjektstrukturen. Zur Konstitutionstheorie des interaktionistischen Konstruktivismus. VS, Wiesbaden, 1999.
- [Tu16a] Tulodziecki, Gerhard: Aktuelle Debatten beim GMK-Forum 2015 im „Rückspiegel“. In (Brüggemann, Marion; Knaus, Thomas; Meister, Dorothee, Hrsg.): *Kommunikationskulturen in digitalen Welten – Konzepte und Strategien der Medienpädagogik und Medienbildung*. kopaed, München, 83-98, 2016.
- [Tu16b] Tulodziecki, Gerhard: Konkurrenz oder Kooperation? Zur Entwicklung des Verhältnisses von Medienbildung und informatischer Bildung. *MedienPädagogik* 25/16, 7-25, 2016, <http://www.medienpaed.com/article/view/428>, 02.05.2017.
- [Tu97] Tulodziecki, Gerhard: *Medien in Erziehung und Bildung*. Klinkhardt, Bad Heilbrunn, 1997.
- [Vo10] Vollbrecht, Ralf; Wegener, Claudia: *Handbuch Mediensozialisation*. VS, Wiesbaden, 2010.
- [Wa16] Wagner, Ulrike; Ring, Sebastian: Organisierte Wildnis – Kooperation von außerschulischer und schulischer Medienpädagogik. In (Knaus, Thomas; Engel, Olga, Hrsg.): *Wi(e)derstände – Digitaler Wandel in Bildungseinrichtungen* (Bd. 5 der fraMediale-Reihe). kopaed, München, 139-149, 2016.
- [We98] Wenger, Etienne: *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, Cambridge (UK), 1998.
- [Wi02] Winkler, Hartmut: Das Modell. Diskurse, Aufschreibesysteme, Technik, Monumente – Entwurf für eine Theorie kultureller Kontinuität. In (Pompe, Hedwig; Scholz, Leander, Hrsg.): *Archivprozesse. Die Kommunikation der Aufbewahrung*. Dumont, Köln, 297-315, 2002.

Lizenz

Dieser Beitrag steht mit dem Einverständnis des Verlags unter folgender Creative Commons Lizenz: CC-BY 3.0 (<https://creativecommons.org/licenses/by/3.0/de/>) und darf unter den Bedingungen dieser freien Lizenz genutzt werden.

Calliope mini - Informatik zum Anfassen für GrundschülerInnen

Stephan Noller¹

Abstract: Im Vortrag wird der Gründer und Erfinder von Calliope mini über die Motivation der Leute hinter dem Projekt sprechen und versuchen die Frage zu beantworten, warum es sinnvoll ist, Kindern schon in der Grundschule das Programmieren beizubringen. Ausserdem wird ausführlich darauf eingegangen, inwiefern mit Calliope mini das Ziel der Förderung von digital literacy erreicht werden kann.

Keywords: Grundschule; Calliope; Digital Literacy

¹ Calliope gemeinnützige GmbH, Liniestraße 52, 10119 Berlin, Deutschland, info@calliope.cc

Vorträge

Zieldimensionen für frühe informatische Bildung im Kindergarten und in der Grundschule

Nadine Bergner¹, Hilde Köster², Johannes Magenheim³, Kathrin Müller³, Ralf Romeike⁴, Ulrik Schroeder¹ und Carsten Schulte³

Abstract: Dieser Artikel stellt die Grundlagen und wesentlichen Ergebnisse einer Expertise für die Stiftung Haus der kleinen Forscher e. V. zum Thema informatische Bildung in Kita und Grundschule vor. Hierzu wurde zunächst eine bildungstheoretisch orientierte Analyse hinsichtlich des Kerns der Fachwissenschaft Informatik durchgeführt. Basierend hierauf wie auch auf weiteren Analysen der Potenziale und Lebensumstände der Kinder sowie entsprechender internationaler Curricula für die Primarstufe werden Kompetenzbereiche zur frühen informatischen Bildung erarbeitet. Dazu wird auf den Rahmen der GI-Standards für die Sekundarstufen zurückgegriffen, die für die Zielgruppe der Kita- und Grundschulkindern um den weiteren Prozessbereich „P0 - Interagieren und Explorieren“ erweitert werden. Anschließend werden aus dem so erweiterten Rahmen vier Kombinationen von Prozess- und Inhaltsbereichen besonders im Hinblick auf deren allgemeinbildende Bedeutung als förderwürdig vorgeschlagen. Ergänzt werden sie um Empfehlungen und Vorschläge für die praktische Umsetzung sowie Ansätze für weiterführende Forschungsfragen.

Keywords: Frühe informatische Bildung, Grundschule, Kindergarten, Kita, Allgemeinbildung

1 Einleitung

Viele Kinder in Deutschland wachsen zurzeit mit großartigen Möglichkeiten und Perspektiven auf. Eine im Alltag wirkmächtige, derzeit aber im Bildungssystem nicht in dem Ausmaß widerspiegelte Ursache sind die Veränderungen durch die Digitalisierung, die derzeit alle Lebensbereiche erfasst und transformiert. Sie führt dazu, dass grundsätzlich Daten aller digitalisierbaren Lebensbereiche zu geringen Kosten maschinell erfasst, gespeichert, verarbeitet, übermittelt und verbreitet werden können. Durch die damit einhergehende enorme Steigerung der Verfügbarkeit von Information ergeben sich weitreichende Möglichkeiten und Herausforderungen für die Gesellschaft.

Derzeit ist noch unklar, wie auf diese umfassenden Veränderungen am besten reagiert werden kann. Die Autoren dieses Beitrags haben in einer Expertise für die Stiftung Haus der kleinen Forscher e. V. (HdkF) anhand theoretischer Überlegungen, den Blick in

¹ RWTH Aachen, Lerntechnologien & Didaktik der Informatik, Ahornstraße 55, 52074 Aachen, [bergner, schroeder]@informatik.rwth-aachen.de

² FU Berlin, Grundschulpädagogik, Habelschwerdter Allee 45, 14195 Berlin, hilde.koester@fu-berlin.de

³ Universität Paderborn, Institut für Informatik, Fürstenallee 11, 33102 Paderborn, [Johannes.Magenheim, Kathrin.Mueller, Carsten.Schulte]@uni-paderborn.de

⁴ FAU Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, ralf.romeike@fau.de

vergleichbare Länder und deren Ansätze sowie der Analyse verschiedener Praxisprojekte und des Forschungsstands in der Didaktik der Informatik untersucht, welchen Beitrag die informatische Bildung zu einer zeitgemäßen und zukunftssicheren Bildung in Kindertagesstätten und Grundschulen leisten kann. Im vorliegenden Beitrag sollen die wichtigsten Erkenntnisse hierzu vorgestellt werden.

Aus unserer vor allem informatikdidaktisch geprägten Sichtweise ist klar: Die neuen digitalen Medien sind nicht nur eine weitere Erscheinungsform im Alltag der Kinder, sie stellen vielmehr einen neuen und eigenständigen Bildungsbereich dar. Zentrales Ziel dieses neuen Bildungsbereichs bzw. dieser neuen Perspektive ist das eigenständige, verantwortungsvolle Handeln in einer digital geprägten Lebenswirklichkeit. Um dieses zu erreichen, bedarf es der Kenntnis der grundlegenden Funktionsprinzipien und Wirkungsweisen digitaler Technologien, da diese – im wahrsten Sinne des Wortes – sonst zwar oberflächlich genutzt, aber nur unzureichend erschlossen, (mit-)gestaltet und bewertet werden können.

2 Analysen der Ausgangslage

2.1 Analyse und Beschreibung der Fachwissenschaft Informatik

Wie ist Informatik darzustellen, um Kindern in der frühen Bildung die notwendigen Grundlagen und ein adäquates Bild zu vermitteln? Informatik wird auch heute noch als die Wissenschaft der automatischen Informationsverarbeitung definiert [GI16]. Sie hat ihre Wurzeln in der Mathematik und den Ingenieurwissenschaften, ähnelt in den Methoden und Fragestellungen aber auch den Naturwissenschaften [TA13].

Wir nehmen vor allem die ingenieurwissenschaftliche Perspektive der Informatik als eine **konstruierende Wissenschaft** auf, die sich im Kern mit Fragen rund um und über die Konstruktion von digitalen Artefakten – Hardware und Software – beschäftigt. Für Bildung sind vor allem die grundsätzlichen Fragen interessant, wie diese Konstruktion im Prinzip vonstattengeht, was das eigentlich ist, was konstruiert wird und wie dadurch das Leben des Einzelnen und der gesamten Gesellschaft beeinflusst wird. Ziel der Informatik ist es zunächst vor allem, effiziente auf digitalen Artefakten ausführbare Algorithmen zu entwickeln, die Abläufe automatisieren und Daten transformieren.

In der Fachdidaktik entwickelte sich rasch ein Konsens, dass das Lösen realweltlicher Probleme im Mittelpunkt stehen sollte. Diese Orientierung wird auch als Modellieren im Gegensatz zum Programmieren bezeichnet. Für die Modellierung – also vor allem die Schritte der Analyse und des Entwurfs – bedeutet das Folgende: Ausgangspunkt ist eine bestimmte (Problem-)Situation. Diese muss zunächst hinsichtlich eines Zwecks abgegrenzt und verallgemeinert werden, denn (nur) verallgemeinerte und damit wiederkehrende Aspekte einer Situation lassen sich sinnvoll automatisieren, weil die Lösung dann auch wiederkehrend eingesetzt werden kann. Entscheidend ist nun der Unterschied zwi-

schen der realen Welt und der Modellwelt. In der realen Welt ist meist intuitiv klar, worum es sich bei einem Gegenstand oder einer Erscheinung handelt, doch die genaue Grenzziehung fällt oft schwer: Wo fängt beispielsweise der Berg an und wo hört das Tal auf? Idealtypen zu beschreiben ist meist leicht, aber es gibt immer wieder Unschärfen. Z.B.: Ab welcher Anzahl nennen wir eine Ansammlung von Bäumen Wald? In der Modell-Welt bzw. der informatischen Welt müssen alle diese Unschärfen auf eindeutige Aussagen bezogen werden können. Dieses Zurückführen ist Grundvoraussetzung für Digitalisierung und Automation. In der informatischen Konstruktion wird also eine Eindeutigkeit hergestellt, die nicht immer so gegeben ist. In der Entwicklung werden hier Entscheidungen getroffen, die oft aus rein softwaretechnischer Sicht relativ egal sind – etwa wie groß das n ist, ab dem man von Wald spricht. Aus Anwenderperspektive kann das durchaus sehr relevant sein, etwa, wenn Eigentümer für ‚Wald‘ einen anderen Steuersatz als für eine Ansammlung von Bäumen bezahlen müssen.

Informatische Lösungen stellen also immer eine Eindeutigkeit her, die es ggf. vorher so nie gegeben hat. Sie tun dies immer zweckbezogen, und dieser Einsatzzweck bedingt eine weitere wichtige Konsequenz: Eine Veränderung der Wirklichkeit, eine Wirkung, ist immer beabsichtigt. Die automatische Ausführung soll etwas neu schaffen oder ersetzen – und damit wird das digitale Artefakt im Einsatz Teil der zuvor analysierten und antizipierten Wirklichkeit. Die Lösung kommt also prinzipiell immer zu spät: Wenn sie eingesetzt wird, gibt es die Situation gar nicht mehr, für die sie einmal entwickelt wurde. Das treibt einen Kreislauf aus sich verändernden Softwareversionen an. Die (Weiter-) Entwicklung von Programmen bzw. digitalen Systemen und Infrastrukturen erfolgt in einer Ko-Evolution: Auf der einen Seite ausgelöst von sich verändernden Bedingungen und neuen Ideen im Einsatzkontext, der sozialen Seite, und auf der anderen Seite ebenso getrieben von inner-technischen Anforderungen und Weiterentwicklungen. Digitale Systeme bzw. Informatiksysteme werden daher zunehmend als soziotechnische Systeme aufgefasst [Ma08].

Für die Grundschule sollte daher ein Weg gefunden werden, diese Konstruktionsprozesse verständlich zu machen. Dazu ist eine Möglichkeit das Gestalten (das nicht mit Programmierung in eins gesetzt werden sollte), eine andere Möglichkeit ist die entsprechende Untersuchung fertiger Systeme, die Exploration (vgl. [Kö06]). In der Expertise [Be17] werden diese beiden Zugangsweisen und ihre Wechselbeziehungen jeweils durch einen Ablauf von Handlungen und Lernprozessen beschrieben (Gestaltungs- und Explorationskreis, vgl. auch [Sc17]).

2.2 Potenziale und Lebensumstände der Kinder

Die Exploration eignet sich auch deshalb, weil Kinder auch in Kita- und Grundschulalter bereits über Nutzungserfahrungen verfügen und diese sinnvoll aufgegriffen werden können. Verschiedene Studien zu Kindern in digitalen Welten belegen, dass Kinder zwar insgesamt selbstsicher mit Informatiksystemen agieren, dabei allerdings häufig die zugrundeliegenden Prinzipien nicht kennen und daher auch die Konsequenzen ihres Han-

delns nicht immer richtig einschätzen können (u.a. [DI17]). Ferner ist ihre Mediennutzung weitgehend konsumierend (Medienangebote im Web nutzen, Videos ansehen, spielen). Sie vernachlässigen das mögliche konstruktive und kreative Potenzial, mit und innerhalb von Informatiksystemen Neues zu schaffen und sich ihre digitalen Welten individuell nützlich zu gestalten. Unserer Einschätzung nach könnte eine besondere Rolle von Schulen und ggf. auch Kindergärten darin bestehen, neben den vorherrschenden konsumierenden Nutzungsaktivitäten weitere Interaktionsmöglichkeiten und -typen zu fördern und aufzuzeigen. Ferner könnten Interaktionen nicht nur isoliert und individuell erlebt werden, sondern Informatiksysteme durch Eltern und Kinder zusammen gestaltet werden, sodass die Erfahrungen dann auch versprachlicht, verarbeitet und altersgerecht reflektiert und eingeordnet werden können.

Untersuchungen zu lernpsychologischen Voraussetzungen sprechen dafür, dass informatische Bildung in Kindertagesstätten und in der Grundschule sowohl sinnvoll als auch möglich ist (bspw. [Sc01] oder [An13]). Die Kinder sind prinzipiell kognitiv dazu in der Lage, ausgewählte informatische Konzepte nachzuvollziehen, zu verstehen und umzusetzen. Sie sind für informatische Aspekte zu begeistern und zwar Mädchen und Jungen gleichermaßen.

Auch wenn informatische Bildung für Kinder im Kindergarten- und Grundschulalter ein relativ junges Forschungsgebiet darstellt, existieren bereits verschiedene erprobte Zugänge, die Kinder motivieren und Kompetenzen der Informatik vermitteln können. Dabei können Zugänge mit und ohne Informatiksystem genutzt werden. Programmierbare Spielzeuge erlauben, Erfahrungen im kreativen, gestalterischen, programmierenden Umgang mit technischen Geräten zu sammeln und diese dabei spielerisch zu erkunden. Alternativ stehen Mikrowelten und Entwicklungsumgebungen mit visuellen, blockorientierten Programmiersprachen zur Verfügung. Diese Alternativen bieten Kindern die Möglichkeit, frühzeitig die Phänomene der durch Informatiksysteme geprägten Welt aktiv zu erkunden und erste Schritte des informatischen Denkens zu erfahren. Hinsichtlich der Eignung für die verschiedenen Zielgruppen lassen sich Zugänge finden, die für Kinder im Kindergartenalter intendiert und erprobt sind (bspw. Lernmaterialien der frühen Bildung zu logischem Denken oder Mathematik, Bücher (z. B. [Liu17]), Bee-Bot, ScratchJr) sowie Angebote für Kinder ab dem Grundschulalter (häufig etwa ab 3. Klasse, bspw. CS Unplugged, Informatik-Biber, LEGO WeDo, Scratch, Schülerlabore). Allerdings ist die Forschungslage hierzu als ungenügend zu bezeichnen, da die verschiedenen Erfahrungsberichte über den Praxiseinsatz allenfalls erste Hinweise zur Gestaltung und Wirkung informatischer Bildung in dieser Altersgruppe geben können.

2.3 Vergleich internationaler Curricula für die Primarstufe

Aktuell etablieren international einige fortschrittliche Bildungssysteme informatische Bildung in der Sekundarstufe I und in der Primarstufe als durchgängige Pflichtfächer, um Schülerinnen und Schülern informatische Denkweisen und Problemlösekompetenzen mittels informatischer Methoden (international „Computational Thinking“) zu vermit-

keln. Diese Denkweisen qualifizieren sowohl für eine fundierte und reflektierte Nutzung von, als auch die kreative Gestaltung der eigenen Lebenswelt mit Informatiksystemen. Daher werden die Fächer allgemein auch mit Begriffen wie Informatik (Computer Science, Informatics, Programming, Computing (UK, NZ)), informatisches Denken (Computational Thinking (USA)) oder Digitale Technologien (Digital Technologies (AUS)) bezeichnet. Unabhängig von ihrer Bezeichnung beinhalten die internationalen Initiativen für frühe informatische Bildung zum einen Bedien- und Medienkompetenz (Digital Literacy und ICT-Grundbildung) sowie zum anderen Problemlösefähigkeiten und technische Gestaltungskompetenzen (Computing, Computational Thinking), die jeweils im Nutzungskontext reflektiert werden müssen.

Die in der Expertise analysierten Kompetenzen aus den internationalen Standards und Curricula wurden in den Ordnungsrahmen der GI-Standards der Sekundarstufen dargestellt. Dabei wurde herausgearbeitet, dass dieser Rahmen weitgehend genutzt werden kann, die Bereiche allerdings unterschiedlich intensiv bzw. mit zielgruppenspezifischen Schwerpunkten zu behandeln sind. Ferner wurde deutlich, dass der gesamte Schnittstellenbereich zur ICT-Anwendungskompetenz und *digital literacy* innerhalb der informatischen Kompetenzen nur abgedeckt werden kann, wenn ein zusätzlicher Prozessbereich definiert wird. Dieser Prozessbereich, den wir im Folgenden als *P0: Interagieren & Explorieren* einführen, lässt sich in den internationalen Standards vielfach finden.

Aus den Kompetenzbeschreibungen der internationalen Ansätze geht häufig nicht direkt hervor, wie sich Kinder mit den Fachinhalten auseinandersetzen. Daher lässt sich eine Zuordnung aller beschriebenen Kompetenzen eher zu den Inhaltsbereichen angeben und dabei die möglichen Prozessbereiche andeuten, da die Art der Auseinandersetzung mit einem Fachinhalt von der didaktischen Auseinandersetzung in der unterrichtlichen Situation abhängt. Häufig sind die Kompetenzen lediglich als „Verstehen“ oder „Erkennen“ ausgewiesen. Das Verständnis können die Kinder aktiv erlangen, indem sie etwas modellieren & implementieren (P1) und dabei Teile begründen & bewerten (P2). Dabei werden oft Aspekte strukturiert & vernetzt (P3) und je nach didaktischer Methode dabei kooperiert & kommuniziert (P4). Fachinhalte müssen auch jeweils dargestellt und die fachliche Darstellung interpretiert werden (P5). Vor allem müssen sich Kinder in der aktiven Auseinandersetzung mit Fachkonzepten, Algorithmen, Programmen oder Informatiksystemen Strukturen und Abläufe erschließen, indem sie mit diesen interagieren und diese explorieren (P0), um sich ein mentales Modell aufzubauen. Daher erweitern die Autoren der Expertise das GI-Kompetenzmodell der Sekundarstufen um den Prozessbereich „P0: Interagieren & Explorieren“.

3 Kompetenzbereiche zur frühen informatischen Bildung

Ausgehend von unseren wissenschaftstheoretischen Überlegungen zur Informatik und daraus resultierenden fachdidaktischen Implikationen, den Erwägungen zur Lebenswelt der Kinder und ihren Erfahrungen mit digitalen Medien sowie unter Berücksichtigung

der Analyseergebnisse internationaler CS-Curricula in der Primarstufe schlagen wir vor, zur Beschreibung von Kompetenzbereichen zur frühen informatischen Bildung das bisher in den Curricula zu den Sekundarstufen I und II verwendete Modell der GI Bildungsstandards [GI16, Be17] um die Prozessdimension „P0: Interagieren und Explorieren“ zu erweitern.

3.1 P0 - Interagieren und Explorieren

Die Exploration eines digitalen Artefakts bzw. eines Informatiksystems ähnelt dem Experimentieren und damit ein wenig dem Forschungskreis der Naturwissenschaften (vgl. [An12], S. 165 pp). Hier geht es darum, anhand der nach außen wahrnehmbaren Funktion bzw. des Einsatzzweckes oder anhand der Nutzung des Systems – also aus der Nutzungsperspektive – zunehmend die innere Bauweise, die einzelnen Bestandteile und deren Wirkmechanismen zu erkunden. Das geht rein durch äußere Beobachtung nur bis zu einem bestimmten Grad, aber die meisten Aspekte der Benutzungsoberfläche sind ja so gestaltet, dass sie die inneren Systemzustände oder Aspekte davon widerspiegeln. Es geht darum, die Aufmerksamkeit weg von der Aufgabe (einen Text verfassen, den Roboter fahren lassen, ein Foto machen) hin zur technischen Realisierung zu lenken (der Text wird nicht bildlich gespeichert, sondern durch Zeichen und Metazeichen, die beschreiben, wie der Text am Bildschirm aussieht...).

Das spielerische Erkunden und Ausprobieren von Objekten ihrer Alltagswelt ist ein wesentliches Element kindlichen Lernens (vgl. [Sa04]). Sie ‚be-greifen‘ derartige Objekte im wörtlichen Sinne und gewinnen über den spielerischen Umgang mit ihnen ein rudimentäres mentales Modell über deren technische Funktionen und ihr mögliches ‚Innenleben‘ (vgl. [SZ97]). Diese lernpsychologischen Erkenntnisse über den kindlichen Umgang mit simplen technischen Artefakten können auch beim Lernen mit und über altersgemäße Informatiksysteme (z. B. Bee-Bot) genutzt werden. Über das Interagieren mit und das Explorieren von Informatiksystemen können sich Kinder einen ersten Zugang zu diesen Systemen erschließen. Wenn es gelingt, diese Erfahrungen in moderierten Lernprozessen für die Kinder zu systematisieren, können die Grundlagen für vertiefte Erkenntnisse der Kinder über Funktion und Struktur derartiger Systeme geschaffen werden.

Neben der lernpsychologischen Begründung dieses Prozessbereiches gibt es auch zahlreiche fachdidaktische und fachwissenschaftliche Argumente, die die Relevanz dieses Prozessbereichs für die informatische Bildung begründen. Zahlreiche internationale Ansätze zur frühen informatischen Bildung, z. B. Resnick am MIT [RR13] oder die CAS-Initiative in Großbritannien, enthalten mit dem ‚Tinkering‘, dem spielerischen Erkunden von Informatiksystemen, ein dem Prozessbereich P0 ‚Interagieren und Explorieren von Informatiksystemen‘ vergleichbares methodisches Konzept.

3.2 Vier hervorzuhebende Kombinationen

Im Folgenden begründen wir ausgewählte Zielkompetenzen auf Ebene der Kinder, die unserer Einschätzung nach die wichtigsten altersgemäßen Verknüpfungen von Prozess- und Inhaltsbereichen demonstrieren. Diese Priorisierung erfolgt anhand von Leitkriterien, die sich sowohl an der fachdidaktischen Relevanz des auszuwählenden Kompetenzbereichs als auch an lern- und entwicklungspsychologischen Kriterien orientieren. Neben den fachlich informatischen Kompetenzen, die sich aus einer Kombination von Inhalts- und Prozessbereichen ergeben und in die vor allem fachdidaktische Überlegungen einfließen, spielt bei der Priorisierung der Aspekt der Vermittlung von allgemeinen Basiskompetenzen und der damit zu erwartende Beitrag zur Allgemeinbildung der Kinder eine wichtige Rolle.

Zusammenfassend liegen die folgenden Leitkriterien der Priorisierung der ausgewählten Kompetenzbereiche zugrunde:

1. **Fachliche und fachdidaktische Konzepte:** Welche Bedeutung hat der ausgewählte Kompetenzbereich für das Verständnis von zentralen informatischen Konzepten und deren Anwendung bei Informatiksystemen?
2. **Lern- und entwicklungspsychologische Aspekte:** Können die informatischen Konzepte auf einem der Zielgruppe angemessenen kognitiven Niveau vermittelt werden?
3. **Alltagsbezug:** Kann der Kompetenzbereich im Hinblick auf die Alltagssituation der Kinder altersgemäß veranschaulicht werden?
4. **Motivation:** Ist der priorisierte Kompetenzbereich anhand eines für die Kinder motivierenden Beispiels darstellbar?
5. **Fachliches Interesse:** Sind die priorisierten Kompetenzbereiche geeignet, das Interesse der Kinder an Informatik zu wecken?
6. **Selbstwirksamkeit:** Befähigt der Kompetenzbereich und das ausgewählte Beispiel die Kinder dazu, ihre Selbstwirksamkeit im Umgang mit Informatiksystemen zu stärken?
7. **Allgemeinbildung:** Können der Kompetenzbereich und das ausgewählte Beispiel einen Beitrag zur Allgemeinbildung der Kinder leisten?
8. **Übergreifende Basiskompetenzen:** Welche Basiskompetenzen, wie Kommunikations- und Kooperationsfähigkeit, Empathie, Problemlösefähigkeit, werden mit dem ausgewählten Kompetenzbereich gefördert?
9. **Bezug zu didaktischen Konzepten** (Grundschule, Sachunterricht, frühkindliche Erziehung): Kann der priorisierte Kompetenzbereich in den curricularen Kontext der Grundschule und in Konzepte der frühkindlichen Erziehung eingeordnet werden?
10. **Existierende Praxiserfahrungen:** Gibt es beim ausgewählten Kompetenzbereich bereits positive Erfahrungen bei der pädagogischen Umsetzung von zielgruppenspezifischen Lernszenarien?

Insgesamt scheinen in Orientierung an diesen Kriterien die folgenden fachlichen Kombinationen aus Inhalts- und Prozessbereichen besonders vielversprechend für eine gelingende informatische Bildung im Elementar- und Primarbereich (vgl. Abb. 1):

- Interaktion mit und Exploration (P0) von Informatiksystemen (I4)
- Modellieren und Implementieren (P1) von Algorithmen und Programmen (I2)
- Repräsentieren/Darstellen/Interpretieren (P5) von Daten und Information (I1)
- Nachdenken über und Bewerten (P2) des Zusammenhangs von Informatik, Mensch und Gesellschaft (I5)

Inhaltsbereiche	(I1) Information & Daten	(I2) Algorithmen & Programmierung	(I3) Sprachen & Automation	(I4) Informatiksysteme	(I5) Informatik, Mensch & Gesellschaft
(P0) Interagieren & Explorieren					
(P1) Modellieren & Implementieren					
(P2) Begründen & Bewerten					
(P3) Strukturieren & Vernetzen					
(P4) Kommunizieren & Kooperieren					
(P5) Darstellen & Interpretieren					

Abb. 1: Empfohlene Kombinationen von Inhalts- und Prozessbereichen. Rot für die Zielgruppe wichtige und naheliegende Kombinationen. Gelb: Weitere zielführende Kombinationen. Beispiele zur Umsetzung werden in der HdkF-Expertise [Be17] ausgeführt.

4 Rahmenbedingungen und Anforderungen für die praktische Umsetzung

Eine erfolgreiche Förderung früher informatischer Bildung hat die Erfüllung verschiedener Bedingungen zur Voraussetzung. Dies betrifft insbesondere die Kompetenzen der pädagogischen Fach- und Lehrkräfte. So sollten pädagogische Fach- und Lehrkräfte (a) fachliche und fachdidaktische Kompetenzen besitzen, insbesondere um angebotenes Lehr-Lern-Material zu verstehen, zu reflektieren und darauf basierend eigene Lerneinheiten zu planen und zu gestalten, (b) pädagogisch-didaktische Handlungskompetenz besitzen, um Lerngelegenheiten bezogen auf informatische Bildung zu erkennen und zu nutzen sowie angebotene Materialien lerngruppenadäquat auszuwählen, anzupassen und

ggf. weiterzuentwickeln, (c) positive Einstellungen zur Informatik aufweisen, aber auch gegenüber existierenden Vorstellungen offen für neue Sichtweisen sein. Die Rahmenbedingungen bezogen auf die Institutionen betreffen vor allem die Schaffung der zeitlichen, räumlichen, organisatorischen und ausstattungsbezogenen Voraussetzungen.

Für den Bereich der informatischen Bildung im Elementar- und Primarbereich liegen bisher keine empirisch fundierten Kompetenzmodelle vor. Daraus resultiert ein Mangel an verlässlichen Instrumenten zur Messung der erworbenen Kompetenzen im Elementar- und Primarbereich der informatischen Bildung und damit zusammenhängend eine empirisch fundierte Evaluation der informatischen Bildungskonzepte im Kita- und Grundschulbereich. Wünschenswert für die weitere Arbeit im Forschungsbereich wäre die konzeptuelle Evaluation entwickelter Maßnahmen und Beispiele, die fachliche Analyse von Materialien und deren Einsatz sowie die Untersuchung der Wirksamkeit in Hinblick auf Kinder, Erzieherinnen, Erzieher und Lehrkräfte, Multiplikatoren und Eltern.

5 Fazit

Die Analyse internationaler Curricula unter fachdidaktischen Gesichtspunkten hat offenbart, dass zusätzlich zu den bestehenden fünf Inhalts- und fünf Prozessbereichen der GI-Bildungsstandards für die Sekundarstufen ein weiterer Prozessbereich „P0 - Interagieren und Explorieren“ notwendig ist, um informatische Bildung bereits im Elementar- und Primarbereich zielgruppenadäquat abzubilden. Durch diesen zusätzlichen Prozessbereich kann die Alltagserfahrung der Kinder zielführend hin zu informatischer Bildung aufgegriffen werden. Die Kinder erlernen anknüpfend an ihre persönlichen Vorerfahrungen die zugrundeliegenden Wirkmechanismen von Informatiksystemen und können so auf zukünftige Herausforderungen der Digitalisierung vorbereitet werden.

Da informatische Bildung im Kita- und Grundschulbereich ein junger Forschungsbereich ist, wurde dieser im zugrundeliegenden Projekt sowohl aus der Perspektive fachdidaktischer Forschung als auch unter Gesichtspunkten pädagogischer Praxis analysiert. Es wird deutlich, dass gut qualifizierte pädagogische Fach- und Lehrkräfte einen wesentlichen Faktor für die erfolgreiche Umsetzung informatischer Bildungskonzepte im Kita- und Grundschulbereich darstellen. Die praktische Erprobung der Beispiele und der vorgestellten Konzepte sowie eine begleitende formative Evaluation werden von den Autoren dabei selbst als wesentliche Faktoren für die erfolgreiche Implementierung früher informatischer Bildung in der Praxis von Kitas und Grundschulen angesehen.

Literaturverzeichnis

- [An13] Anders, Y. et al: Wissenschaftliche Untersuchung zur Arbeit der Stiftung „Haus der kleinen Forscher“, Band 5, Schubi Lernmedien 2013.
- [An12] Ansari, S. et al: Wissenschaftliche Untersuchung zur Arbeit der Stiftung „Haus der kleinen Forscher“, Band 4, Schubi Lernmedien 2012.
- [DI17] Kinder in der digitalen Welt. (o. J.). DIVSI: <https://www.divsi.de/publikationen/studien/divsi-u9-studie-kinder-der-digitalen-welt/>, Stand: 14.02.2017.
- [GI16] GI, Gesellschaft für Informatik e.V.: Bildungsstandards Informatik für die Sekundarstufe II, Empfehlungen der Gesellschaft für Informatik e.V., Arbeitskreis „Bildungsstandards SII“: LOG IN, 2016.
- [Be17] Bergner, N., Köster H., Magenheimer, J., Müller, K., Romeike, R., Schulte, C.; Schroeder, U.: Zieldimensionen informatischer Bildung im Elementar- und Primarbereich. In (Stiftung Haus der kleinen Forscher, Hrsg.): Frühe informatische Bildung – Ziele und Gelingensbedingungen für den Elementar- und Primarbereich. Berlin, 2017 (im Druck)
- [Kö06] Köster, H.: Freies Explorieren und Experimentieren: eine Untersuchung zur selbstbestimmten Gewinnung von Erfahrungen mit physikalischen Phänomenen im Sachunterricht, Studien zum Physik- und Chemielernen. Berlin : Logos-Verlag, 2006.
- [Li17] Liukas, L.: Hello Ruby: Programmier dir deine Welt. Bananenblau, 2017.
- [Ma08] Magenheimer, J.: Systemorientierte Didaktik der Informatik Sozio-technische Informatiksysteme als Unterrichtsgegenstand? In (Kortenkamp; U. et al, Hrsg.): Informatische Ideen im Mathematikunterricht. Franzbecker, Hildesheim, S. 17-36, 2008.
- [RR13] Resnick, M.; Rosenbaum, E.: Designing for Tinkerability. In (Honey, M.; Kanter, D., Hrsg.): Design, Make, Play: Growing the Next Generation of STEM Innovators, Routledge, New York, S. 163-181, 2013
- [Sa04] Sachser, N.: Neugier, Spiel Und Lernen: Verhaltensbiologische Anmerkungen Zur Kindheit. Zeitschrift Für Pädagogik 50 4, S. 475–86, 2004.
- [Sc17] Schulte, C. et al: The Design and Exploration Cycle as a R&D Framework in CSE in Proceedings of the IEEE EDOCON 2017, Athens, Greece, S. 862 – 871.
- [Sc01] Schwill, A.: Ab wann kann man mit Kindern Informatik machen? In (Keil-Slawik, R.; Magenheimer, J., Hrsg.): INFOS 2001 – 9. GI-Fachtagung Informatik und Schule, S. 13–30, 2001.
- [SZ97] Schwarzkopf, H.; Zolg, M.: Kann der Computer denken? Grundschulzeitschrift, H. 108, 1997.
- [TA13] Tedre, M.; Apiola, M.: Three computing Traditions in school computing education. In: Kadujevich, D.M.; Angeli, C.; Schulte, C.: Improving Computer Science Education, Routledge, New York, 2013.

Was Grundschul Kinder über Informatik wissen und was sie wissen wollen

Sandra-Jasmin Petrut¹, Nadine Bergner¹ und Ulrik Schroeder¹

Abstract: Beim Einstieg in die Informatik in der Grundschule stellen sich die Fragen, wo die Lernenden abgeholt werden müssen, welche Erfahrungen sie mitbringen und was die Lernenden an dem Thema Informatik interessiert und nicht zuletzt motiviert. Im Rahmen eines eintägigen Grundschulmoduls wurden für eine quantitative Analyse 198 Kinder mit Hilfe eines Pre-Post-Tests und für eine tiefere qualitative Analyse 34 Kinder mit der Methode des Plakaterstellens evaluiert. Die Auswertung beider Studien ergab ein konsistentes Ergebnis. Der Computer wird bei Grundschulkindern als „Spielzeug“ statt als Werkzeug bzw. Lernmedium gesehen. Außerdem hat die betrachtete Zielgruppe keine Vorbehalte der Disziplin gegenüber, sondern zeigt ohne geschlechtsspezifische Unterschiede großes Interesse. Dies spiegelt sich auch in den Fragen wider, die die Schülerinnen und Schüler auf die Plakate schrieben.

Keywords: Digitale Bildung, Primarstufe, Evaluation, Schülerlabor, Vorkenntnisse, Bild der Informatik

1 Einleitung und Motivation

Außerschulische Angebote bieten Kindern im Grundschulalter eine Möglichkeit zum Erwerb digitaler Bildung. Neben hardware- und computergestützten Materialien, wie dem Calliope Mini², der Internetseite code.org oder kindgerechten Robotern, gibt es auch technikfreie Varianten, wie beispielsweise Computer Science Unplugged³ der University of Canterbury. Auch Schülerlabore, wie das InfoSphere⁴ der RWTH Aachen, bieten eine Reihe solcher Möglichkeiten. Außerdem gibt es in NRW das Projekt „Informatik an Grundschulen“⁵, welches von den Universitäten Aachen, Paderborn und Wuppertal in Kooperation mit dem Ministerium für Schule und Weiterbildung NRW ins Leben gerufen wurde, um informatische Bildung in den Sachunterricht zu integrieren. Im Kernlehrplan für den Sachunterricht an Grundschulen in NRW [Mi08] ist ein Ziel, einen „verantwortungsvollen Umgang mit der natürlichen und gestalteten Lebenswelt“ [ebd. S. 39] zu fördern. Auch sollen die vielfältigen Erfahrungen der Schülerinnen und Schüler in ihrer Lebenswirklichkeit sowie die Möglichkeit zur Informationsbeschaffung durch „alte“ und „neue“ Medien genutzt werden. Betrachtet man die KIM-Studie für die

¹ RWTH Aachen, Lehr- und Forschungsgebiet Informatik 9, Ahornstr. 55, 52074 Aachen, nachname@informatik.rwth-aachen.de

² calliope.cc

³ csunplugged.org

⁴ schuelerlabor.informatik.rwth-aachen.de

⁵ https://www.schulministerium.nrw.de/docs/Schulsystem/Unterricht/Lernbereiche-und-Faecher/Mathematik-_Naturwissenschaften/Informatik-an-Grundschulen/index.html

Altersgruppe 6 bis 13 aus dem Jahre 2014 [Me15], so erkennt man, dass digitale Medien in beinahe allen Haushalten vorzufinden sind. So ist die heutige Lebenswirklichkeit der Kinder voller Informatiksysteme. Auch die aktuell in Entwicklung befindlichen Empfehlungen für Bildungsstandards Informatik speziell für den Primarbereich sehen eine Anknüpfung an die Lebenswelt der Lernenden im Inhaltsbereich "Informatik, Mensch und Gesellschaft" explizit vor [GI17].

2 Related Work

Dieses Paper lehnt an eine Interviewstudie von Borowski, Diethelm und Wilken aus 2011 an [BDW16]. Dabei wurden über 600 Grundschul Kinder zwischen acht und zehn Jahren aus 30 verschiedenen Klassen interviewt. Die Kinder formulierten Fragen über die Themen Computer, Handys, Roboter etc., die sie einem Experten stellen wollen. Insgesamt wurden 2594 Fragen gesammelt, die die Autoren in die sechs Hauptkategorien: „Internet“ (656), „Computer“ (647), „Roboter“ (450), „Handys“ (405), „Musik und Bild“ (210) und „Spielekonsolen“ (209) zusammenfassten. Außerdem wurden Unterkategorien gebildet, die die Art der Fragen beschreiben. Diese sind „Geschichte und Zukunft“, „Arbeitsweise“, „Potential“, „Entwicklung und Produktion“ und „Sicherheit und Haltbarkeit“.

Die Bedeutung des Fragenstellens für den Lernprozess wurde in zahlreichen internationalen Arbeiten herausgearbeitet und unter anderem von Chin und Osborne 2008 zusammengefasst [CO08]. Das Stellen von Fragen seitens der Schülerinnen und Schüler bietet sowohl für die Lernenden als auch für die Lehrenden einige Vorteile. Auf der einen Seite fördert es das direkte Lernen eines Inhaltes, auf der anderen Seite fällt den jeweiligen Lehrkräften eine Diagnose über die Vorkenntnisse der Schülerinnen und Schüler leichter.

3 Forschungsfragen und -design

3.1 Forschungsfragen

In der hier präsentierten Studie sind Kinder im Grundschulalter (aus NRW) die Zielgruppe. Aufgrund des extrem heterogenen Vorwissens, welches hauptsächlich privat erworben werden kann, ergeben sich folgende Forschungsfragen:

- Was verbinden Grundschul Kinder mit dem Begriff „Informatik“?
- Welches Interesse haben Grundschul Kinder an Informatik?
- Was möchten Grundschul Kinder über Informatik wissen?
- Welche Informatiksysteme nutzen Grundschul Kinder?

3.2 Forschungsdesign

Das Informatik-Schülerlabor InfoSphere bietet für diese Studie den Forschungskontext. Im InfoSphere gibt es verschiedene Module für Kinder und Jugendliche ab der dritten Klasse. Für Grundschul Kinder gibt es das Modul „Zauberschule Informatik“, welches in dieser Studie betrachtet wird. Dieses Modul besteht aus verschiedenen Stationen, welche den Kindern Thematiken aus der Informatik näherbringen. Dabei wird in zwei Stationen der Computer als Werkzeug eingesetzt, an den anderen vier Stationen arbeiten die Kinder mit Papier, Stiften und weiteren Hands-On-Materialien.

In Station 1 „Magische Zahlen“ erfahren die Kinder, dass und wie man jede Dezimalzahl in eine Binärzahl umwandeln kann. In Station 2 „Zettelzauber“ lernen die Kinder die Funktion von Prüfbits kennen und wie man einen Fehler korrigieren kann. In der dritten Station „Türme von Hanoi“ erlernen die Kinder das Prinzip der Rekursion. In Station 4 „Bilder verzaubern“ werden auch Bilder binär dargestellt. In der fünften Station „Verhexte Wege“ arbeiten die Kinder am Computer und versuchen den kürzesten Weg von einem Ort zum nächsten zu finden. So erfahren sie, wie ein Navigationsalgorithmus funktioniert. In der letzten Station „Der Computer“ schauen die Kinder ein Video über den Aufbau des Computers. Anschließend wird ein Zuordnungsquiz bearbeitet, welches die wichtigsten Bestandteile des Computers aufgreift. Für eine nähere Beschreibung der Stationen und des Moduls kann Bergners Dissertation betrachtet werden [Be15].

Um den obigen Forschungsfragen nachzugehen, wird eine Kombination aus einem quantitativen und einem qualitativen Forschungsansatz gewählt. Zum einen werden die Grundschulmodule seit 2012 in einem Pre-Post-Testdesign schriftlich evaluiert, zum anderen wurde die Methode des Plakaterstellens angewandt, um weitere Details zur aktuellen Lebenssituation der Kinder und ihren Interessen zu erheben. Gleichzeitig wird erforscht, inwiefern sich diese kreative Methode als Forschungsinstrument für die Zielgruppe der Grundschul Kinder eignet, um deren Vorstellungen zu erheben.

In der Vorbefragung wird das Vorwissen der Kinder über Informatik wie auch der Besitz und die Nutzung des Computers erhoben. Weiter werden die Teilnehmerinnen und Teilnehmer aufgefordert, ihr Interesse an verschiedenen Aspekten der Informatik anzugeben. In der Nachevaluation wird zum einen ein Feedback zum Modul und die Passung zu den vorherigen Erwartungen erfragt, zum anderen werden die Fragen zum Interesse an verschiedenen Aspekten der Informatik wiederholt.

Die Auswertung geschieht in einem ersten Schritt summativ, um einen Überblick über die Angaben der Schülerinnen und Schüler zu erhalten. In einem zweiten Schritt werden gruppenspezifische Unterschiede (z. B. zwischen Jungen und Mädchen) mittels des (Pearson-)Chi-Quadrat-Tests ermittelt. Ebenso werden abschließend die Veränderungen durch den eintägigen Modulbesuch über den Vergleich der Pre- und Posttests errechnet.

Für die qualitative Evaluation wurden zwei Durchführungen des Moduls „Zauberschule Informatik“ betrachtet. Daran nahmen zwei Grundschulklassen mit je 17 Schülerinnen und Schülern der vierten Klasse teil. Zu Beginn wurden die Kinder in Gruppen von je 4-

5 Schülerinnen und Schüler aufgeteilt. Mit jeweils einem Interviewer saß jede Gruppe um ein Plakat mit der Zeichnung eines Computers. Jedes Kind erhielt einen Stift mit unterschiedlicher Farbe, damit im Anschluss die Beiträge eindeutig einer Person zugeordnet und damit teilgruppenspezifisch ausgewertet werden konnten.

Der Interviewer stellte vorkonzipierte Fragen, die die Schülerinnen und Schüler auf dem Plakat beantworten. Sie hatten dabei die Möglichkeit, einzelne Begriffe oder Sätze zu notieren, aber auch Bilder zu malen. Die gestellten Fragen gehören zu zwei Kategorien. Im ersten Fragenblock ging es um das Vorwissen der Kinder und im zweiten Block wurde nach den Interessen gefragt. Die erste Frage lautete „Was fällt dir zu dem Begriff „Informatik“ ein?“, um zu ermitteln, welche Vorerfahrungen die Kinder mitbrachten. Das Bild des Computers gab den Kindern einen Gedankenanstoß, da vermieden werden sollte, dass ein Kind nichts auf das Plakat schreibt. Auf Grundlage ihrer Antworten wurden vier weitere Fragen gestellt:

1. An welchen Orten/wo könnte Informatik vorkommen?
2. Was könnte zu Hause/in der Schule/draußen mit Informatik zu tun haben?
3. Wo nutzt du einen Computer, ein Smartphone etc.?
4. Mit wem nutzt du den Computer, das Smartphone etc.?

Sowohl die dritte als auch die vierte Frage war davon abhängig, was die Kinder bereits angegeben hatten. Falls beispielsweise kein Kind am Tisch ein Smartphone genannt hatte, so wurde nicht nach dem Smartphone gefragt, um keine Antworten zu erzwingen. Anschließend ging es weiter mit dem zweiten Fragenblock. Hier war die erste Frage „Was interessiert dich an Informatik?“. Auch hier wurden weitere Fragen abhängig der bereits genannten Antworten gestellt:

1. Was interessiert dich an Computern, Smartphones etc.?
2. Was wolltest du schon immer über Informatik wissen?

Zu Beginn der Plakatevaluation stellte sich sowohl jedes Kind als auch der Interviewer vor und schrieben ihre Namen auf das Plakat. So konnte am Ende zwischen Mädchen und Jungen differenziert werden. Mittels der kurzen Vorstellungsrunde und der Tatsache, dass auch der Interviewer seinen Namen auf das Plakat schrieb, sollte das Gefühl des „Getestetwerdens“ vermieden werden.

Um eine möglichst objektive Erhebung zu ermöglichen, stellte der Interviewer ausschließlich die vorkonzipierten Fragen. Auch eine Wertung der Antworten wurde vermieden. Bei Rückfragen der Kinder, ob ihre Antworten richtig seien, wurde ihnen gesagt, dass sie dies alles gleich bei der Moduldurchführung erfahren. Abgesehen von der Freiheit ihrer Nennungen wurde den Kindern zu Beginn mitgeteilt, dass auf Rechtschreibung, Grammatik und künstlerische Begabung nicht geachtet werde, um mögliche Hemmungen zu nehmen.

4 Auswertung der Evaluation

4.1 Quantitative Auswertung

Im Rahmen der quantitativen Evaluation beantworteten im Zeitraum von 2012 bis 2016 insgesamt 223 Kinder den Vor- und Nach-Fragebogen. Dabei ergab sich durch die Teilnahme von Geschwisterkindern an bestimmten Veranstaltungen ein Spektrum der ersten bis siebten Klassenstufe. Für die folgende Auswertung wurden die Ausreißer herausgefiltert, so dass sich ein Stichprobenumfang von $N=198$ Kindern ergab, von denen 83 Kinder die dritte und 115 Kinder die vierte Klasse besuchten. Damit ergibt sich folgende Altersverteilung: 44 Acht-, 96 Neun-, 52 Zehn- und 6 Elfjährige. Die Stichprobe umfasst 84 Mädchen und 113 Jungen.

Der erste Fragenblock thematisiert die Vorerfahrungen der Kinder mit technischen Geräten und insbesondere mit dem Computer. 87,4% der Besucherinnen und Besucher gaben an, Zugang zu einem Computer zu haben. Diese 173 Kinder wurden tiefergehend dazu befragt, wozu sie den Computer nutzen. In Abb. 1 sieht man die Verteilung in vier Kategorien. Bezüglich der Kategorien „Hausaufgaben machen“ und „Spielen“ konnte eine geschlechtsspezifische Differenz festgestellt werden: Jungen gaben häufiger an, den Computer für die Hausaufgaben bzw. für das Spielen zu verwenden. Diese Bestandsaufnahme macht deutlich, dass der private Computer hauptsächlich als Spielzeug und selten als Lernmedium verstanden wird.

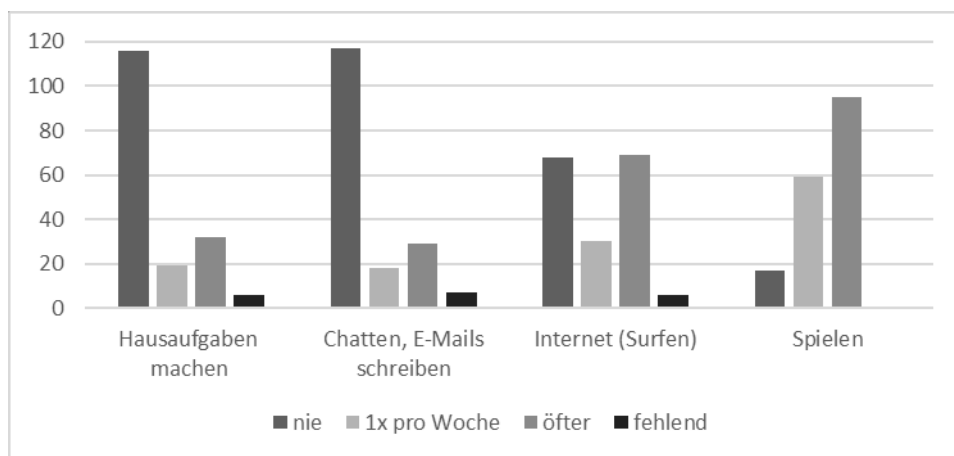


Abb. 1: Nennungen zur Verwendung des Computers

Als Nächstes wurde das grundsätzliche Interesse der Kinder an dem bevorstehenden Modul im Schülerlabor erhoben. Die Auswertung zeigt, dass die wenigsten Kinder der Informatik Vorbehalte entgegenbringen. Auch ergaben sich keine geschlechtsspezifischen Differenzen. Im Vergleich des Pre- und Posttests wurden die Erwartungen der Besucherinnen und Besucher mit der Erfüllung jener verglichen. 38 Kinder notierten

„viel Spaß“. Dies zeigt, dass die Kinder keine inhaltlichen Erwartungen an den Tag im Informatik-Schülerlabor hatten. Obwohl die Angaben zu den Erwartungen im Vorfeld vage blieben, gaben 115 Kinder nach dem Modul an, dass jenes ihren Erwartungen entsprochen habe, 23 Kinder stimmten dem nicht zu. Zusätzlich gab ein Großteil der Kinder an „großes Interesse“ zu haben einen weiteren Kurs zu besuchen, welches wiederum keine signifikanten Differenzen zwischen den Geschlechtern, den Kindern mit und ohne Computerzugang und der Klassenstufe aufzeigt.

Bezogen auf das vorhandene Bild zur Informatik zeigt sich vor dem Modulbesuch, dass die Vorstellung kaum ausgeprägt ist (Tabelle 1). Der Vergleich macht deutlich, dass der Bezug zu den Informatiksystemen auch in den Unplugged Modulen deutlich wird.

Kategorie	Anzahl Nennungen im Pretest	Anzahl Nennungen im Posttest
Kein Begriff	52	21
Computer	30	41
Technik	10	15

Tab. 1: Nennungen zu Informatik

Das Interesse der Kinder an verschiedenen Aspekten wurde in vier Kategorien unterteilt (Abb. 2). Die Auswertung zeigte außerdem, dass Jungen größeres Interesse an technischen Geräten haben, umgekehrt scheint das Interesse am Lösen kniffliger Aufgaben bei Mädchen stärker ausgebildet zu sein. Auch an der Informatik zeigten die meisten Kinder starkes Interesse, obwohl sie gleichzeitig kundtaten, dass sie gar nicht wüssten, was dies sei.

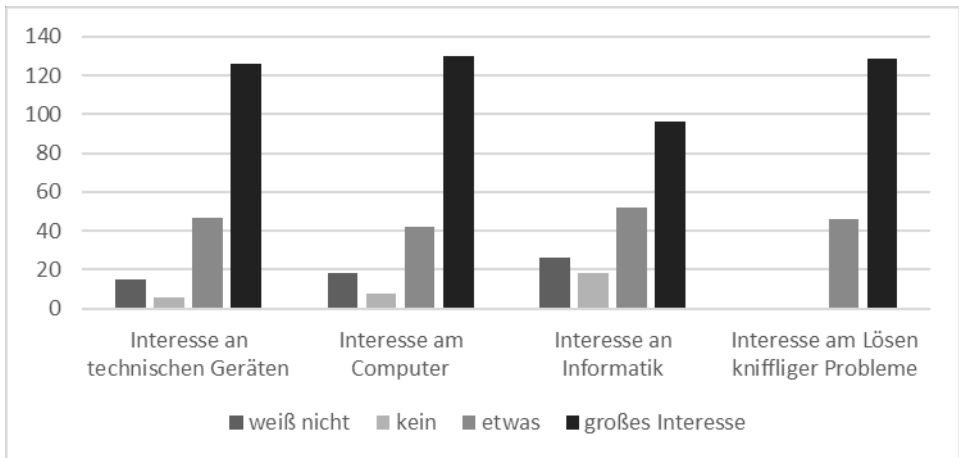


Abb. 2: Interesse an verschiedenen Aspekten

Nach denselben vier Kategorien wurden die Kinder auch im Posttest befragt. Es zeigte sich in allen Kategorien ein Zuwachs des Interesses, wobei dieser zur Informatik am

deutlichsten ausfiel, was allerdings darin begründet ist, dass in den anderen beiden Kategorien die Kinder bereits im Pretest sehr großes Interesse angaben.

4.2 Qualitative Auswertung

Die erstellten Plakate wurden nach jeder Durchführung eingesammelt, anonymisiert und zusammengefasst. Bei den zwei Durchführungen entstanden acht Plakate, an denen insgesamt 34 Schülerinnen und Schüler (13 Mädchen und 21 Jungen) mitgewirkt hatten. Die Kinder waren alle in der 4. Jahrgangsstufe und besuchten dieselbe Schule. Bei der Analyse der Plakate wurden zunächst alle Einzelnennungen, die kompletten Sätze, Fragen und Bilder kategorisiert, wobei die Rechtschreibung korrigiert wurde. Auf Basis der Vorarbeit von Borowski, Diethelm und Wilken wurden zunächst dieselben Hauptkategorien gewählt. Da einige Angaben der Kinder nicht in die vorhandenen Kategorien passten, entstanden weitere Kategorien und bestehende wurden erweitert. Eine Aufzählung der meist genannten Kategorien und zugehöriger Anzahl der Nennungen findet sich in Tabelle 2 in den ersten beiden Spalten.

Kategorie	Anzahl Nennungen	Anzahl Kinder	Mädchen	Jungen	Anzahl Plakate
Spiele	84	28	10 (77%)	18 (86%)	8
Computer	82	28	11 (85%)	17 (81%)	8
Internet	57	27	10 (77%)	17 (81%)	8
Bild, Musik, Video	46	25	8 (62%)	17 (81%)	8
Kommunikationsmittel	39	26	11 (85%)	15 (71%)	8
Tablet	24	20	8 (62%)	12 (57%)	7
Schule	22	14	6 (46%)	8 (38%)	5
Programmierung	15	11	5 (38%)	6 (29%)	6

Tab. 2: Kategorienverteilung

Aus der Kategorie „Bild und Musik“ wurde „Bild, Musik, Video“, die Kategorien „Spielekonsolen“ und „Spiele“ wurden zu „Spiele“ zusammengefasst und aus der Kategorie „Handy“ entstand die Kategorie „Kommunikationsmittel“.

Die Anzahl der Nennungen überschreitet die Anzahl der Kinder, zumal Nennungen in den Einzelbegriffen, aus kompletten Sätzen, aus den Fragen oder den Bildern hinzugezählt wurden. Darüber hinaus ist in Tabelle 2 die Anzahl der Mädchen und der Jungen angegeben, die mindestens eine Nennung zu den jeweiligen Kategorien hatten. Die letzte Spalte zeigt, welche Kategorien auf wie vielen der Plakate vorzufinden waren.

Bereits hierbei lassen sich Unterschiede zu den Ergebnissen von Borowski, Diethelm und Wilken erkennen. Das Thema „Spiele“ steht in der hier dargestellten Studie an erster Stelle, wohingegen die Begriffe „Spielekonsole“ und „Spiele“ in der damaligen Studie insgesamt, also nach dem Zusammenfügen beider Kategorien zum Aspekt „Spiele“, nur auf dem fünften Platz lag. Ein Unterschied zwischen Mädchen und Jungen wird bei der Auswertung nicht erkennbar. Dies kann unter anderem daran liegen, dass an allen

Plakaten sowohl Schülerinnen als auch Schüler arbeiteten, die sich gegenseitig beeinflusst haben könnten.

Die gestellten Fragen wurden neben der Einteilung in die Hauptkategorien auch in die Unterkategorien „Potential“ (Po, 24x), „Entwicklung und Produktion“ (EP, 21x), „Arbeitsweise“ (AW, 8x), „Vergangenheit und Zukunft“ (VZ, 3x) und „Sicherheit und Haltbarkeit“ (SH, 1x) zugeordnet. Insgesamt wurden 57 Fragen von 27 Kindern gestellt.

In Tabelle 3 werden Fragen der Hauptkategorien „Spiele“, „Computer“, „Internet“ und „Musik, Bild, Video“ den zugehörigen Unterkategorien zugeordnet. Da jedoch nicht jede Frage eindeutig einer Haupt- und einer Unterkategorie zugeordnet werden kann, wiederholen sich die Fragen oder gehören der Unterkategorie „Sonstiges“ an.

UK ⁶	Fragen	HK ⁷
Po	Hat man bei Informatik meistens einen Computer?	Computer
	Gehört Informatik zum Computer?	
	Für was braucht man einen Fernseher?	Musik, Bild, Video
EP	Wie schaffen es Programmierer Spiele zu machen?	Spiele
	Wie programmiert man ein Videospiel?	
	Wie wurden (alle) Spiele hergestellt?	
	Wer hat den Computer erschaffen?	Computer
	Wie wird ein Computer gebaut?	
	Wer baut Computer?	
	Wie entsteht WLAN?	
Wie wurden Pixel hergestellt?	Musik, Bild, Video	
Wie wurden Filme auf Fernseher hergestellt?		
AW	Wie funktioniert eine elektrische Dartscheibe?	Spiele
	Was kann man auf dem Computer alles machen?	Computer und Musik, Bild, Video
	Kann man auf dem Computer Fotos machen?	
	Kann man auch Fotos und Videos machen?	
Wie funktionieren Online-Server?	Internet	
VZ	Seit wann gibt es PS1, PS2, PS3, PS4 und Wii?	Spiele
	Seit wann gibt es Computer?	Computer
	Können Computer bald Menschen ersetzen?	
SH	Wie geht man mit dem Computer um?	Computer
Sonstiges	Wie viele Computer gibt es auf der Welt?	Computer
	Wie viele Computerapps gibt es auf dem Computer?	
	Wie viele Internetseiten gibt es?	Internet
	Kann man in Bilder reinspringen und in eine andere Realität kommen?	Musik, Bild, Video
	Kann man Pixel in die Realität setzen?	

Tab. 3: Unterkategorienverteilung

⁶ Unterkategorie

⁷ Hauptkategorie

Die Fragen zum Ort des Vorkommens und der Nutzung von Informatiksystemen wie auch den damit verbundenen Personen konnten nicht ausgewertet werden. Fast alle Kinder beantworteten diese drei Fragen nur mündlich. Dies könnte daran gelegen haben, dass ihnen nicht eindeutig bewusst war, wie sie ihre Antworten auf das Plakat eintragen konnten. Außerdem wurde die erste Frage von einigen Kindern auch missverstanden, sodass sie Geschäfte nannten, wo Informatiksysteme zu kaufen sind (z.B. Media Markt, Saturn, Handyladen etc.). Die Auswertung der letzten Frage ergab kein Ergebnis. Alle Kinder, die hierauf antworteten, gaben an, dass sie entweder mit ihren Eltern oder mit ihren Geschwistern die zu Hause befindlichen Geräte nutzen.

5 Reflexion und Ausblick

Zunächst zeigt sowohl die quantitative als auch die qualitative Auswertung, dass die befragten Kinder (ebenso wie in den Ergebnissen der KIM-Studie) häufig den Computer nutzen. Auch in der Auszählung der Beiträge auf den Plakaten ist die Kategorie „Computer“ auf Platz 2 gelandet. Jedoch sollte hier vermerkt werden, dass die Kinder den Computer im Wesentlichen als „Spielzeug“ sehen. Dies bestätigt sich durch die quantitative Auswertung der Tätigkeiten, welche die Kinder am Computer ausführen. Bei näherer Betrachtung der Nennungen auf den Plakaten zeigt sich, dass häufig explizite Computerspiele genannt wurden, so wird ebenfalls ein Zusammenhang zwischen dem Computer und den Spielen sichtbar. Insgesamt sind keine geschlechtsspezifischen Differenzen erkennbar. Dies wird zum einen in Hinblick auf die Interessen der Schülerinnen und Schüler und zum anderen auch bei ihren Vorbehalten der Disziplin gegenüber deutlich.

Aus der quantitativen Auswertung ergibt sich, dass über ein Viertel der Kinder kein Vorwissen über Informatik haben. Dies lässt nicht nur darauf schließen, dass es keinen Informatikunterricht in der Grundschule gibt, sondern auch, dass in anderen Fächern kein Bezug zur Informatik hergestellt wird. Auch ergibt sich die Vermutung, dass Kinder den Begriff „Informatik“ nicht zuordnen können, obwohl sie dennoch Vorwissen mitbringen. Hier wird für zukünftige Evaluationen nach Alternativen gesucht ohne mit anderen Assoziationen Einfluss auf die Nennungen zu nehmen. Außerdem lag die Kategorie „Schule“ in der Plakatevaluation mit nur 22 Nennungen weit hinten. Dieses Ergebnis ist verwunderlich, da die jeweiligen Lehrkräfte mitteilten, dass die Kinder den „Internetführerschein“ in der Schule machten. So könnte man schlussfolgern, dass die Anzahl der Nennung der Kategorie „Schule“ bei anderen Grundschulklassen noch geringer ausfallen könnte.

Aus der Auswertung der von den Kindern notierten Fragen wird erkennbar, dass sie großes Interesse an den Unterkategorien „Potential“ und „Entwicklung und Produktion“ haben. Diese Fragen können ohne spezifisches Vorwissen beantwortet werden, indem danach recherchiert wird. Dies könnte eine Motivation sein, erste Aspekte informatischer Bildung bereits in der Grundschule zu vermitteln, um die Interessen der Kinder

abzudecken, ohne hochqualifizierte Informatiklehrkräfte in der Grundschule zu benötigen. Jedoch sollte nicht außer Acht gelassen werden, dass weitere informatische Aspekte, die ebenfalls das Interesse der Kinder wecken und für eine informatische Grundbildung zwingend notwendig sind, nur von qualifizierten Lehrkräften vermittelt werden können (z. B. das Thema Internet).

Abschließend kann festgehalten werden, dass die Evaluationsmethode „Plakat erstellen“ gelungen ist, sodass sie in weiteren Durchführungen erneut eingesetzt wird. Zusätzlich wird intensiv in Richtung Vorbilder geforscht. So ist hier die Frage offen, mit welchen Personen die Kinder Informatiksysteme assoziieren bzw. mit wem sie diese verwenden und welche Rolle die Berufe der Eltern oder auch die Informatikerfahrung von Geschwistern und Freunden spielen. Aus diesem Grund wird auch in dem Interviewleitfaden die Frage „Wem würdest du deine Fragen stellen?“ ergänzt. Bei weiteren Durchführungen werden auch die Lehrkräfte hinzugezogen, um die aktuelle Situation und das mögliche Vorwissen der Schülerinnen und Schüler zu erheben. Dabei würde sich ein Interview mit der jeweiligen Lehrkraft beim Besuch im InfoSphere eignen. Darüber hinaus wird ein Beobachtungsbogen für die Betreuerinnen und Betreuer angefertigt, sodass in zukünftigen Durchführungen auch die Aktivität der Kinder sowie weitere nicht verschriftliche Beiträge der Diskussion ausgewertet werden können.

Literaturverzeichnis

- [Be15] Bergner, N.: Konzeption eines Informatik-Schülerlabors und Erforschung dessen Effekte auf das Bild der Informatik bei Kindern und Jugendlichen. Dissertation, Lehr- und Forschungsgebiet Informatik 9, RWTH Aachen, 2015.
- [BDW16] Borowski, C.; Diethelm, I.; Wilken, H.: What children ask about computers, the Internet, robots, mobiles, games, etc.. ACM-Press, S. 72-75, Münster, 2016.
- [CO08] Chin, C.; Osborne, J.: Students' question: a potential resource for teaching science. In Studies in Science Education 44.1. S. 1-39, 2008.
- [Me15] Medienpädagogischer Forschungsverbund Südwest: KIM-Studie 2014 – Kinder + Medien, Computer + Internet. Stuttgart, 2015.
- [Mi08] Ministerium für Schule und Weiterbildung des Landes NRW: Richtlinien und Lehrpläne für die Grundschule in Nordrhein-Westfalen: Deutsch, Sachunterricht, Mathematik, Englisch, Musik, Sport, Evangelische Religionslehre, Katholische Religionslehre. Ritterbach, Frechen, 2008.
- [GI17] Arbeitskreis „Bildungsstandards Primarbereich“ der Gesellschaft für Informatik e.V.: Bildungsstandards Informatik für den Primarbereich – zur Diskussion. <http://metager.to/gibspdf>, Stand: 02.01.2017.

Algorithmisieren im Grundschulalter

Sabrina Hoffmann¹, Katharina Wendlandt², Matthias Wendlandt³

Abstract: Informatik, insbesondere das Programmieren, wird zunehmend in Form von Projekten in Grundschulen in Deutschland thematisiert. Eine der wesentlichen Schlüsselkompetenzen zum erfolgreichen Programmieren ist das Algorithmisieren von Problemlösungen. In der vorliegenden Arbeit wird zunächst ein Stufenmodell des Algorithmisierens entworfen und darauf aufbauend ein Testverfahren zur Beurteilung der Fähigkeit zum Algorithmisieren gewonnen. Basierend auf diesem Testverfahren wurden 155 Grundschul Kinder aus acht Grundschulklassen der Schulstufen 2-4 bezüglich ihrer Kompetenz zum Algorithmisieren untersucht. Die zugrundeliegende Fragestellung war, inwiefern sich die Fähigkeit zum Algorithmisieren in den Schulstufen 2-4 entwickelt. Mithilfe des Exakten Fisher-Tests konnten signifikante Unterschiede in vier verschiedenen Stufen der Fähigkeit zum Algorithmisieren zwischen den Klassenstufen nachgewiesen werden. Dabei fiel auf, dass der Unterschied zwischen der zweiten und dritten Klasse deutlicher ausfiel als zwischen der dritten und vierten Klasse. Die vorliegende Arbeit stellt den ersten Teil einer zweigeteilten Untersuchung dar. Der zweite Teil der Untersuchung wird sich mit der Trainierbarkeit des Algorithmisierens im Grundschulalter beschäftigen.

1 Einleitung

Informatikunterricht trägt in vielfältiger Weise zur kognitiven Entwicklung, der persönlichen Entfaltung und zu einem selbstbestimmten Handeln eines jeden Schülers/ einer jeden Schülerin bei. Informatische Kompetenzen wie das Algorithmisieren, das Modellieren und die Problemlösefähigkeit bilden heutzutage grundlegende Fähigkeiten, die in vielen Situationen auch unabhängig von Informatik verwendet werden können. Notwendiges Wissen über Datenschutz, Datensicherheit und Grenzen des Computers befähigt die SuS dazu, ein selbstbestimmtes Leben zu führen. Deshalb ist es eine Kernfrage, ob und inwieweit Informatik schon in der Grundschule angeboten werden kann. Diese Frage wurde bereits in einigen Projekten angegangen.

Mit der Zauberschule [BLS11] zeigen die Autoren einige interessante Ansätze zum Thematisieren von Informatik in der Grundschule. Die Themen unterteilten sich in Binärzahlen, Bilddarstellung und Fehlererkennung. Die Autoren berichten, dass es den Kindern viel Spaß gemacht habe. In [Sc01] konnte aus psychologischen Studien gefolgert werden, dass Grundschul Kinder in der Lage seien, gewisse fundamentale Ideen der Informatik altersgerecht zu verstehen und anzuwenden. In [BD09] wird ein Projekt im Rahmen einer Arbeitsgemeinschaft mit Kindern der vierten Klasse beschrieben, welches das Thema Programmieren durch die drei Programmierumgebungen *Karol*, *Scratch* und *Lego Mindstoms* thematisiert. Die Reflexion des Projekts zeigt, dass *Karol* und *Lego Mindstoms* für einen

¹ Rotebergschule Dillenburg, Rühlstraße 3, 35683 Dillenburg, Deutschland

² Mathematisches Institut, Universität Giessen, Arndtstr. 2, 35392 Giessen, Deutschland

³ Institut für Informatik, Universität Giessen, Arndtstr. 2, 35392 Giessen, Deutschland

solchen Kurs weniger geeignet sind, wohingegen *Scratch* eine gute Möglichkeit darstellt, Kinder dieser Altersstufe erste Programmiererfahrungen sammeln zu lassen. Ein weiteres Projekt, welches das Programmieren in Scratch [GW16] in der Grundschule thematisiert, wurde von der Hochschule Hannover in einer zweiten Grundschulklasse durchgeführt. Auch hierbei kommen die Autoren zu dem Schluß, dass „Grundschul Kinder der zweiten Klasse in der Lage sind, Programmierkonzepte zu erlernen“.

Die oben genannten Publikationen, bzw. Projekte, resultieren alle in dem Ergebnis, dass Grundschul Kinder in der Lage sind zu programmieren. Diese Aussage erscheint gerade in der zweiten Klasse überraschend, da entwicklungspsychologische Aussagen dies nicht vermuten lassen würden und es stellt sich die Frage, *inwieweit* Grundschul Kinder fähig sind zu programmieren. Nach Untersuchungen von Piaget [SL12] sind Kinder im präoperationalen Stadium (2-7 Jahre) nicht imstande, geistige Operationen vollständig durchzuführen. Diese Fähigkeit entwickelt sich erst im Verlauf des konkret-operationalen Stadiums (7-12 Jahre). Da das Grundschulalter die Zeit zwischen dem 6. und dem 10. Lebensjahr umfasst, sind grundlegende geistige Entwicklungsprozesse im Gange. Auch Untersuchungen mathematischer Kompetenzen im Grundschulalter bestätigen den Entwicklungsprozess. Erklärungen finden sich in der Fähigkeit Repräsentationsformen beziehungsweise Methoden zu kennen und nutzen zu können, beziehungsweise in den vorhandenen Fähigkeiten des Arbeitsgedächtnisses [RHP08, Gr06].

Die vorliegende Arbeit geht der Frage nach, auf welchem Komplexitätsniveau Grundschüler programmieren können und welche Entwicklungsphasen dabei sichtbar werden. Den Schwerpunkt der Untersuchung bildet dabei die Fähigkeit des Algorithmisierens. Das Algorithmisieren ist eine der wesentlichen überfachlichen Kompetenzen, die im Informatikunterricht vermittelt werden. Ein Algorithmus ist eine präzise Folge von Anweisungen zum Lösen eines genau definierten Problems. Während die Kompetenz Algorithmen anzuwenden oft eine sehr spezielle und auf ein Fach oder eine Wissenschaft bezogene Tätigkeit ist, ist die Kompetenz des Algorithmisierens, also die Fähigkeit für ein gegebenes Problem eine präzise Folge von Lösungsschritten zu finden, eine Kompetenz, die in vielfältiger Weise, in verschiedenen Wissenschaften und vor allem im Alltag eine wichtige Rolle spielt. Sie ist die Fähigkeit, eine Problemlösung präzise und zielgerichtet zu konstruieren. Darüber hinaus liegt es im Charakter des Fachs Informatik, eine möglichst effiziente, also in möglichst wenig Zeit und mit möglichst wenig Platz, ablaufende Problemlösung zu beschreiben, die darüber hinaus noch automatisiert werden kann. Damit grenzt sich die Informatik deutlich von anderen Wissenschaften ab. Die Algorithmisierung findet sich deshalb auch in jeglicher Strukturierung der Informatik aus schulischer Sicht wieder (vgl. z.B. [Bi16], [Sc93]).

In Kapitel 2 wird zunächst ein Stufenmodell des Algorithmisierens entwickelt, welches die drei Dimensionen *Programmier-elemente*, *Raumkomplexität* und *Problemkomplexität* umfasst. Die einzelnen Dimensionen werden jeweils in Stufen unterteilt. Aus diesem Stufenmodell werden in einem zweiten Schritt Testaufgaben entwickelt, um die Fähigkeiten zum Algorithmisieren von insgesamt 155 Grundschulern aus acht Klassen der Klassenstufen 2-4 an drei verschiedenen Schulen in Hessen bzw. Rheinland-Pfalz zu untersuchen. Die Aufgaben beziehen sich auf die Programmierung eines Hamsters [Bo13], der Aufga-

ben in einem Territorium bewätigen muss. Wie in den eingangs erwähnten Projekten wird auch hier mit konkret sichtbaren Problemstellungen gearbeitet, um die Aufgaben für die SuS leichter verständlich zu machen und die Motivation zu erhöhen. Die Ergebnisse befinden sich in Kapitel 4. Die dazugehörige Diskussion und ein Ausblick auf zukünftige Ideen befindet sich in Kapitel 5.

2 Das Stufenmodell

Zur Klassifizierung der Programmierfähigkeiten wurden Dimensionen des Programmierens [MHB15, BR12] untersucht. Auch die Taxonomie von Biggs [BC82] kann als Maß adaptiert werden. Jedoch sind beide Modelle zu grobkörnig für die Untersuchung der vorliegenden Problemstellung, da das Spektrum der Dimensionen für die Fähigkeiten von Grundschulern zu umfassend ist. Im Folgenden wird Algorithmisierung in drei verschiedene Dimensionen unterteilt. Die erste Dimension beschreibt die Möglichkeiten Algorithmen zu beschreiben. Hierbei werden drei verschiedene Stufen unterschieden:

1. Sequenzielle Befehle
2. Alternativen
3. Iteration

Als grundlegende Programmierelemente kommen normalerweise noch *Methoden* und in objektorientierten Programmiersprachen *Objekte* hinzu. Beide zählen wir nicht zur Algorithmisierung, sondern ähnlich wie auch in [Sc93] zur Modellierung, da sie die Funktionalität nicht erweitern, jedoch die Struktur verändern. *Rekursion* könnte als eigenständige Stufe betrachtet werden. Sie kann jedoch immer durch Iteration ersetzt werden und erweitert den Beschreibungsbereich somit nicht. Darüber hinaus ist die Rekursion ein fortgeschrittenes Konzept der Beschreibung von Lösungen, welches nur selten im Alltag intuitiv verwendet wird. Aus diesem Grund ist Rekursion für eine Einführung in der Grundschule sicherlich nicht geeignet. In höheren Klassenstufen kann sie jedoch als eigenständige Stufe mit aufgenommen werden.

Die zweite Dimension bezieht sich auf den Problemraum. Bei der gewählten Umgebung ist dies das Territorium des Hamsters (vgl. Bild 1). So kann ein Problem einen fest vorgegebenen oder einen beliebigen Raum betreffen. Der Algorithmus kann somit nur für genau einen Raum funktionieren oder für eine ganze Klasse von Territorien. Bei einem fest vorgegebenen Raum wird dahingehend unterschieden, ob der Weg fest vorgegeben ist oder nicht:

1. vorgegebener Raum, Weg vorgegeben
2. vorgegebener Raum
3. beliebiger Raum

Bei Problemen mit einem vorgegebenem Weg müssen die Kinder nicht selbst einen Weg finden, sondern dieser ist durch die Beschaffenheit des Raums vorgegeben (vgl. Bild 4 im Gegensatz zu Bild 5). Probleme in einem beliebigen Raum bedürfen der Fähigkeit, Lösungen nicht nur an konkreten Situationen festzumachen, sondern ein Muster oder etwas Vergleichbares zu erkennen. Ein Beispiel hierfür wäre „Beschreiben Sie einen Algorithmus, der Sie aus einem beliebigen Labyrinth führt“. Nach Piaget [SL12] ist diese Fähigkeit, also die Fähigkeit, Verhalten unabhängig von einer konkreten Situation geistig zu verarbeiten, im konkret-operationalen Stadium (7-12 Jahre) noch nicht herausgebildet. Deshalb kann diese Stufe nur als Ausblick gewertet werden. Sie wird im Folgenden nicht untersucht.

Die dritte Dimension umfasst die Problemkomplexität. Hierbei werden ebenfalls drei verschiedene Stufen unterschieden:

1. 2-3 Änderungen
2. 4-8 Änderungen
3. beliebig viele Änderungen, Erkennung und Bearbeitung von Mustern

Untersuchungen zur Verarbeitungs- und Behaltensspanne [PSZ11] zeigen, dass 6-jährige Kinder dazu fähig sind, etwa 4 Wörter unmittelbar reproduzieren zu können. Die Verarbeitungsleistung ist noch geringer einzuschätzen, vgl. [Gr06]. Nach der Durchführung von Vortests mit Zweit- und Drittklässlern zeigte sich, dass Probleme mit vier oder mehr Änderungen, die sich die SuS merken mussten, deutlich schlechter gelöst wurden. Die letzte Stufe (beliebig viele Änderungen) kann nur in Kombination mit der dritten Stufe der Dimension *Problemraum*, also *beliebiger Raum*, auftreten. Deshalb wird sie in dieser Arbeit nicht untersucht.

Die unterschiedlichen Kombinationen von Komplexitätsstufen werden im Folgenden an einigen Stellen mit Zahlen abgekürzt. So beschreibt die Bezeichnung 3.2.1 ein Problem, dass unter Verwendung von Iteration, in einem vorgegebenen Raum mit maximal 2-3 Änderungen gelöst werden muss.

3 Methode

Insgesamt wurden von den SuS sieben Aufgaben bearbeitet. Alle Aufgaben wurden mit Zettel und Stift bearbeitet. Es wurde bewusst auf den Einsatz von Computern verzichtet, da sich bei der Verwendung von Computern die Möglichkeit bietet, eine richtige Lösung durch Austesten zu gewinnen. Dies misst jedoch nicht die Fähigkeit zum Algorithmisieren.

Bevor die konkreten Aufgaben beschrieben und analysiert werden, sei erwähnt, dass die Untersuchung nacheinander drei Befehlssätze verwendet hat. Befehlssatz 1 wurde in den Aufgaben 1-4 verwendet. Befehlssatz 2 in den Aufgaben 2-4. Für die Aufgaben 5-7 wurde ausschließlich Befehlssatz 3 verwendet.

Befehlssatz 1 wurde folgendermaßen definiert: **vor** - Er geht ein Kästchen vor, **links** - Er geht ein Kästchen nach links, **rechts** - Er geht ein Kästchen nach rechts, **zurück** - Er geht ein Kästchen zurück.

Befehlssatz 2 umfasst die Befehle: **vor bis zur Wand** - Er geht vor bis zur Wand, **links bis zur Wand** - Er geht nach links bis zur Wand, **rechts bis zur Wand** - Er geht nach rechts bis zur Wand, **zurück bis zur Wand** - Er geht zurück bis zur Wand.

Befehlssatz 3 umfasst die Befehle: **vor** - Er geht ein Kästchen vor, **linksum** - Er dreht sich nach links (90 Grad), **rechtsum** - Er dreht sich nach rechts (90 Grad), **Wand** - Er geht vor bis zur Wand.

Im Folgenden werden die Aufgaben erklärt. Zunächst die ersten beiden Aufgaben und Aufgabe 5:

1. Das ist unser Hamster Emil. Er hat einen neuen Käfig bekommen und kennt sich noch nicht so gut aus. Er wacht nachts auf und hat Hunger. Leider ist es sehr dunkel und er sieht nichts. Bitte hilf ihm, damit er das Futter findet (siehe Bild 1).
2. Emil will sein neues Spielzeug, einen Tunnel, ausprobieren. Er ist noch ein bisschen ängstlich. Du musst ihn führen. Du darfst kein Kommando zweimal hintereinander benutzen (vor vor ist nicht erlaubt!) (siehe Bild 2).
5. Emil hat Futter gefunden und möchte es in seinen Bau bringen. Bitte hilf ihm! Leite ihn zum Futter und dann zurück zum Ausgangspunkt.



Bild 1. Aufgabe 1

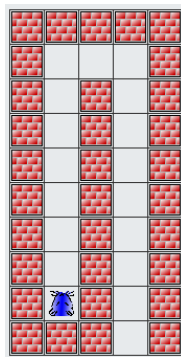


Bild 2. Aufgabe 2

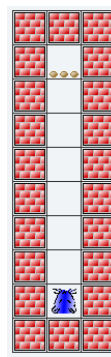


Bild 3. Aufgabe 5

Diese Aufgaben dienen zum einen als Einführung der Befehlssätze und „Warming up“, zum anderen sollen sie zeigen, dass die SuS die Befehle verstanden haben, wissen, wie sie ihre Lösungen aufschreiben sollen und prinzipiell keine links-rechts Schwächen haben. Sie zielen noch nicht darauf ab, eine der Stufen der Komplexität (vgl. Kapitel 2) zu messen.

Der erste Block von bewerteten Aufgaben (3+4) beschäftigt sich mit der Stufe der *Iterationen*. Hierbei wird die erste Stufe des Problemraums (*vorgegebener Raum, vorgegebener*

Weg) in Aufgabe 3 und die zweite Raumstufe in Aufgabe 4 (*vorgegebener Raum*) getestet. Die Problemkomplexität bleibt in der untersten Kategorie, da die Tests vor allem Aussagen über die Fähigkeit des Konzepts der Iteration geben sollen. Es werden also Stufe 3.1.1 und Stufe 3.2.1 getestet.

Die Aufgaben 3 und 4:

3. Puh! Der Tunnel ist ja noch gar nicht zu Ende und jetzt geht er auch noch im Zick-Zack weiter. Führe Emil durch den Tunnel.
Emil, mache solange
...
bis du aus dem Tunnel kommst!
4. Vom vielen Laufen hat Emil aber Hunger. Benutze möglichst wenig Anweisungen.
Emil, mache solange
...
bis du oben an der Wand bist. Gehe nun noch
...

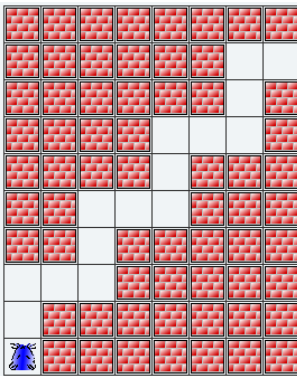


Bild 4. Aufgabe 3

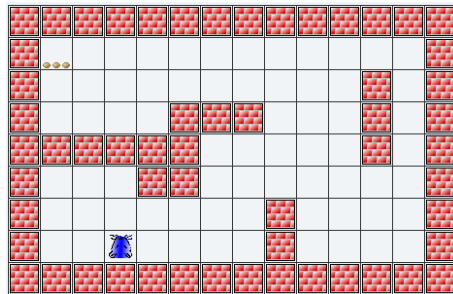


Bild 5. Aufgabe 4

In den Aufgaben 6 und 7 wurde die Problemkomplexität untersucht. So wurden bei beiden Aufgaben die ersten beiden Stufen auf unterstem Niveau betrachtet (*sequentielle Befehle*, *vorgegebener Raum*, *vorgegebener Weg*). Aufgabe 6 misst die Problemkomplexität auf Stufe 1 (2-3 Änderungen) und Aufgabe 7 misst die Problemkomplexität auf Stufe 2 (4-8 Änderungen). In Vortests zeigte sich, dass sich die Kinder Positionen im Raum durch Benutzung des Fingers auf dem Aufgabenzettel merkten. Deshalb wurde bei den Aufgaben 5-7 die Drehung aufgenommen und darauf geachtet, dass die SuS nur einen Stift zur Lösung benutzen konnten. Auf diese Weise konnte sichergestellt werden, dass die SuS sich gewisse Situationen merken mussten. Diese Fähigkeit wird beim Programmieren häufig benötigt, zum Beispiel bei Veränderungen von Variablen oder Arrays in einem Programm.

Die Aufgaben 6 und 7:

- 6 Emil ist entlaufen und ist in ein Rohr gekrabbelt. Er hat Angst und möchte deshalb nur noch vorwärts laufen, weil er dort alles gut sieht. Könntest du ihm helfen, wieder heraus zu kommen?
- 7 Emil findet wieder einmal das Futter nicht. Führe Emil zu den 6 Honigbällchen.

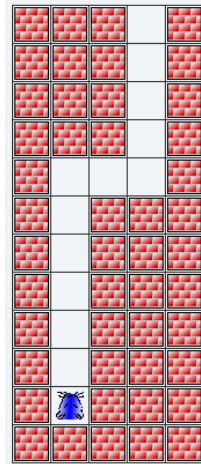


Bild 6. Aufgabe 6

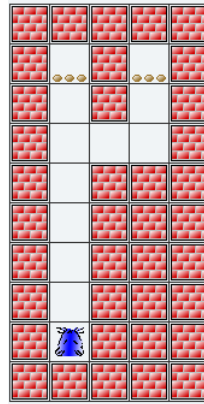


Bild 7. Aufgabe 7

Um die Objektivität der Untersuchung zu gewährleisten, wurde ein einheitlicher Ablaufplan für alle Untersuchungen aufgestellt. Sowohl die Erklärungen der Befehle, als auch die gezeigten Beispiele und der organisatorische Ablauf waren somit einheitlich und klar vorgegeben. Darüber hinaus wurden alle Untersuchungen unter Anleitung derselben Person durchgeführt.

4 Ergebnisse

Die bearbeiteten Aufgaben wurden bewertet und die Resultate in die diskrete Ordinalskala bestehend aus den drei Merkmalen „richtig gelöst“, „teilweise gelöst“ und „nicht gelöst“ eingeordnet. Die Bewertung „teilweise gelöst“ wurde vergeben, wenn kleinere Fehler gemacht wurden, aber die Lösung des Problems erkennbar war. Syntaxfehler, bzw. Rechtschreibfehler, wurden nicht als Fehler gewertet. Die Ergebnisse der einzelnen Aufgaben sind in Kontingenztafeln dargestellt.

Die statistische Auswertung der Ergebnisse wurde mittels des *Exakten Fisher-Tests* durchgeführt. Die Berechnung erfolgte durch das Statistikprogramm *R*. Der Nachweis der Leistungsunterschiede zwischen den einzelnen Klassenstufen erfolgt zum Signifikanzniveau 1%. Die nachgewiesenen Unterschiede gelten als *hoch signifikant*.

Sind Unterschiede in den Leistungen zwischen der zweiten und dritten und zwischen der dritten und vierten Klasse nachweisbar, so kann hier aufgrund der Daten transitiv auf einen Leistungsunterschied zwischen den Klassen 2 und 4 geschlossen werden. Nur in Einzelfällen werden daher die Ergebnisse der zweiten und vierten Klasse direkt miteinander verglichen.

Aufgabe 3	Klasse 2	Klasse 3	Klasse 4	
richtig gelöst	5	31	39	75
teilweise gelöst	1	3	0	4
nicht gelöst	27	32	17	76
	33	66	56	155

Tabelle 1. Ergebnisse Aufgabe 3

Bei Aufgabe 3 ist zwischen der zweiten und dritten Klasse ein signifikanter Unterschied nachweisbar ($p = 0.002$). Auch wenn im Vergleich zu den SuS aus Klasse 3 ein höherer Prozentsatz der SuS aus Klasse 4 die Aufgabe richtig lösen konnte, ist dieser Unterschied nicht signifikant ($p = 0.014$). Ein deutlicher Leistungsunterschied besteht außerdem zwischen der zweiten und vierten Klasse ($p < 0.001$).

Aufgabe 4	Klasse 2	Klasse 3	Klasse 4	
richtig gelöst	1	16	29	46
teilweise gelöst	0	5	0	5
nicht gelöst	32	45	27	104
	33	66	56	155

Tabelle 2. Ergebnisse Aufgabe 4

Bei Aufgabe 4, der Aufgabe auf der höheren Problemstufe 3.2.1, lassen sich signifikante Unterschiede zwischen der zweiten und dritten Klasse ($p = 0.003$) und zwischen der dritten und vierten Klasse ($p = 0.001$) nachweisen. Es kann außerdem festgestellt werden, dass nur ein Kind aus der zweiten Klasse die Aufgabe lösen konnte.

Aufgabe 6	Klasse 2	Klasse 3	Klasse 4	
richtig gelöst	10	29	31	70
teilweise gelöst	7	17	14	38
nicht gelöst	16	20	11	47
	33	66	56	155

Tabelle 3. Ergebnisse Aufgabe 6

Bei Aufgabe 6 lässt sich kein signifikanter Unterschied zwischen den einzelnen Klassenstufen feststellen. Auch wenn angenommen werden muss, dass die SuS der zweiten und dritten Klasse die Aufgabe gleichermaßen lösen konnten ($p = 0.216$), genauso wie die SuS

der dritten und vierten Klasse ($p = 0.350$), so zeichnet sich doch ein Unterschied zwischen den SuS der zweiten und vierten Klasse ab ($p = 0.015$).

Aufgabe 7	Klasse 2	Klasse 3	Klasse 4	
richtig gelöst	6	34	38	78
teilweise gelöst	8	13	6	27
nicht gelöst	19	19	12	50
	33	66	56	155

Tabelle 4. Ergebnisse Aufgabe 7

Bei der höheren Problemkomplexität in Aufgabe 7 ist ein signifikanter Unterschied zwischen der zweiten und dritten Klasse nachweisbar ($p = 0.003$). Der Unterschied in den Ergebnissen der dritten und vierten Klasse ist nicht signifikant ($p = 0.184$). Es liegt ein deutlicher Unterschied zwischen den Leistungen der SuS der zweiten und vierten Klasse vor ($p < 0.001$).

5 Diskussion und Ausblick

Es zeigt sich ein deutlicher Leistungsunterschied zwischen der zweiten und der dritten Klasse. So konnten in den Aufgaben 3, 4 und 7 signifikante Unterschiede nachgewiesen werden. Auffällig ist auch, dass, bis auf eine einzige Ausnahme, kein Schüler und keine Schülerin der zweiten Klasse das Konzept der Iteration umsetzen konnte. Hingegen zeigt sich in der dritten Klasse, dass nahezu die Hälfte der Kinder unmittelbar fähig waren, dieses Konzept umzusetzen, sofern der Weg vorgegeben war. Bei nicht vorgegebenem Weg waren es nur noch ein Viertel der Drittklässler, die die Aufgabe lösen konnten. Der Unterschied zwischen zweiter und dritter Klasse wird auch in Aufgabe 7 sehr deutlich. So nimmt die Leistung bei einer höheren Anzahl von Veränderungen, die verarbeitet werden müssen, stark ab.

Der Unterschied zwischen den Klassen 3 und 4 zeigt sich vor allem bei dem Konzept der Iteration. So ist der Unterschied bei Aufgabe 3 mit vorgegebenem Weg zwar vorhanden, aber nicht signifikant. Bei Aufgabe 4 hingegen haben zirka ein Viertel der Drittklässler die Aufgabe korrekt gelöst, wohingegen mehr als die Hälfte der Viertklässler diese Aufgabe erfolgreich lösen konnte. Die sichtbaren Unterschiede in den Ergebnissen der Dritt- und Viertklässler in Aufgabe 7 müssen als zufällig angesehen werden.

Es zeigt sich, dass Aufgaben auf sehr niedrigem Niveau in allen Klassenstufen erfolgreich bearbeitet werden konnten. Sowohl das Konzept der Iteration als auch höhere Problemkomplexitätsstufen führten zu großen Problemen bei Schülerinnen und Schülern der zweiten Klasse. Das Konzept der Iteration kann unmittelbar erst ab der dritten Klasse verstanden und angewendet werden. Der sichere Umgang mit Iteration kann erst ab der vierten Klasse erwartet werden.

Insgesamt zeigt die Untersuchung klare Unterschiede im Übergang zwischen der zweiten und der dritten Klasse. Hinzu kommt, dass das Konzept der Iteration eine deutliche

Hürde in allen untersuchten Jahrgangsstufen ist. In einer Folgeuntersuchung muss überprüft werden, inwieweit diese Stufen in den betrachteten Altersklassen trainierbar sind. So sollte nach einem Eingangstest eine Unterrichtseinheit oder Ähnliches mit Grundschulkindern durchgeführt werden und infolgedessen ein zweiter Test durchgeführt werden. Hierbei muss vor allem die Verwendung des Programmierelements *Iteration* in allen Jahrgangsstufen untersucht werden. Desweiteren müssen die Bedingungen der zweiten Klasse genauer untersucht werden, da hier die prinzipielle Frage ist, ob Schülerinnen und Schüler der zweiten Klasse fähig sind, Lösungen für Probleme auf höheren Stufen zu erstellen. Aber auch für die Klassen 3 und 4 stellt sich die Frage, ob sich die Fähigkeiten für eine größere Anzahl von Schülerinnen und Schüler in einer Unterrichtseinheit erlernen, bzw. festigen lassen. Für die vierte Klasse stellt sich darüber hinaus die Frage, ob die Schülerinnen und Schüler schon in diesem Alter fähig sind, Probleme in einem beliebigen Raum zu lösen, da dies in klassischen Programmieraufgaben eine wichtige Rolle spielt.

Literatur

- [BC82] John B. Biggs, Kevin F. Collis, *Evaluating the quality of learning: The SOLO taxonomy; structure of observed learning outcome*, Educ. Psychology Series, Academic Pr., 1982.
- [BD09] Christian Borowski, Ira Diethelm, *Kinder auf dem Weg zur Informatik: Programmieren in der Grundschule*, In: Zukunft braucht Herkunft, INFOS 2009, 244–253, 2009.
- [Bi16] Gerhard Röhner et al., *Bildungsstandards Informatik für die Sekundarstufe II*, Beilage zu LOG IN, Heft Nr. 183/184, 2016.
- [BLS11] Nadine Bergner, Thiemo Leonhardt, Ulrik Schroeder *Zauberschule Informatik – Einblick in die Welt der Informatik für Kinder im Grundschulalter*, In: Informatik mit Kopf, Herz und Hand, INFOS 2011, 132–141, 2011.
- [Bo13] Dietrich Boles, *Programmieren spielend gelernt mit dem Java-Hamster-Modell*, Springer-Vieweg-Verlag, 2013.
- [BR12] Karen Brennan and Mitchel Resnick, *New frameworks for studying and assessing the development of computational thinking*, In: AERA, 2012.
- [Gr06] Dietmar Grube, *Entwicklung des Rechnens im Grundschulalter*, Waxmann, 2006.
- [GW16] Robert Garmann, Benjamin Wanous, *Code for competence - Programmieren für Zweitklässler mit ScratchJr*, In: Informatik für Kinder - 7. Münsteraner Workshop zur Schul-informatik, 71–80, 2016.
- [MHB15] Andreas Mühlring, Peter Hubwieser, Marc Benges, *Dimensions of programming knowledge.*, In: Situation, Evolution, and Perspectives, International Conference on Informatics in Schools, 32–44, Springer, 2015.
- [PSZ11] Martin Pinquart, Gudrun Schwarzer, Peter Zimmermann, *Entwicklungspsychologie - Kindes- und Jugendalter*, 111–121, Hogrefe Verlag, 2011.
- [RHP08] Kristina Reiss, Aiso Heinze, Reinhard Pekrun, *Mathematische Kompetenz und ihre Entwicklung in der Grundschule*, In: Kompetenzdiagnostik, 107–127, Verlag für Sozialwissenschaften, 2008.
- [Sc01] Andreas Schwill, *Ab wann kann man mit Kindern Informatik machen?*, In: Informatikunterricht und Medienbildung, INFOS 2001, 13–30, 2001.
- [Sc93] Andreas Schwill, *Fundamentale Ideen der Informatik*, Zentralblatt für Didaktik der Mathematik 1, 20–31, 1993.
- [SL12] Wolfgang Schneider, Ulman Lindernberger *Entwicklungspsychologie*, 385–400, Beltz Verlag, 2012.

Bild der Informatik von Grundschullehrpersonen

Erste Zwischenergebnisse aus qualitativen Einzelfallstudien

Alexander Best¹

Abstract: Über eine schriftliche Umfrage an Grundschulen im Regierungsbezirk Münster (NRW) konnten Lehrpersonen für Einzelinterviews über ihre Vorstellungen zur Informatik und zu Informatikunterricht gewonnen werden. Die Lehrpersonen assoziieren Gegenstände, Inhalte und Methoden der Informatik, der Medienbildung, des unterrichtlichen Medieneinsatzes sowie bestehender Grundschulfächer mit Informatikunterricht. Sie stellen Bezüge zu verschiedenen Inhalts- und Prozessbereichen der Empfehlungen für Bildungsstandards Informatik für die Sekundarstufe I der Gesellschaft für Informatik (GI) her. Kindern den kreativen, kritischen, sachgerechten und zielgerichteten Umgang mit Informatiksystemen zu vermitteln, wird von den Interviewten als wesentliches Ziel informatischer Bildung in der Grundschule verstanden. Es werden Zwischenergebnisse des Forschungsvorhabens auf Basis von zwei Interviews vorgestellt.

Keywords: Informatik in der Grundschule; Primarbildung; Bild der Informatik; Lehrervorstellungen; Einzelfallstudien; Leitfadeninterviews; Qualitative Forschung

1 Einleitung

Die Forderungen nach informatischer Bildung im Primarbereich haben in den vergangenen Jahren stark zugenommen [ACM12]. Als möglicher Lehr-Lernort wurde, neben außerschulischen Angeboten, der Klassenunterricht an Grundschulen genannt (ex. [BDM10]). Bislang liegen kaum gesicherte Erkenntnisse über Vorstellungen von Grundschullehrpersonen zur Informatik und zu Informatikunterricht in Deutschland vor [BM15, FGH16]. Diese stellen eine wesentliche Voraussetzung dar, um Unterrichtskonzepte sowie Aus-, Fort- und Weiterbildungsangebote in einem *Bottom-Up*-Verfahren entwickeln zu können [DKW11]. Für den Sekundar- und Tertiärbereich existieren bereits Arbeiten zu diesem Forschungsfeld (ex. [Be15]).

Im Rahmen des Forschungsprojekts Informatik in der Grundschule (IGS) am Arbeitsbereich Didaktik der Informatik der Westfälischen Wilhelms-Universität Münster werden Studien zu diesem Desiderat durchgeführt. Im Projekt werden u.a. folgende Forschungsfragen untersucht:

- Welche biografischen Bezüge zur Informatik und zu Informatikunterricht werden von Grundschullehrpersonen assoziiert?

¹ Westfälische Wilhelms-Universität Münster, Didaktik der Informatik, Fliegenerstr. 21, 48149 Münster, alexander.best@uni-muenster.de

- Wo begegnen Grundschul Kinder in ihrem Alltag nach Ansicht von Grundschullehrpersonen Informatik?
- Wie sind zentrale Begriffe der Informatik und des Informatikunterrichts nach Auffassung von Grundschullehrpersonen inhaltlich belegt und welche Bedeutung messen sie diesen Begriffen der Primarbildung bei?

2 Methodologie und Methodik

Im Forschungsprojekt IGS wird ein hypothesengenerierender Ansatz im Kontext qualitativer Forschung verfolgt. Eine erste Pilotstudie im Jahr 2015 zeigte, dass über die Erhebung quantitativer Daten zwecks Hypothesenüberprüfung nur ein sehr eindimensionaler Zugang zum Forschungsfeld geschaffen werden konnte [BM15]. Erste Zwischenergebnisse aus den bislang geführten Einzelinterviews lassen vermuten, dass den Lehrpersonen durch die methodische Neuausrichtung mehr Raum für die Darlegung ihrer individuellen Perspektive gegeben wird.

Zur Datenerhebung wurde ein Interviewleitfaden nach dem sogenannten SPSS-Prinzip entwickelt [He11]. Es wurden zunächst Fragen des Forschers gesammelt (S), anhand von Kriterien geprüft (P), sortiert (S) und subsummiert (S). Der so entstandene Interviewleitfaden wurde einer befreundeten Grundschullehrperson zur Prüfung vorgelegt, um unverständliche Leitfragen bzw. Fragen der *Check*-Liste zu identifizieren und umzuformulieren. Anschließend erfolgte ein erstes Testinterview, bei dem sämtliche Fragen erneut auf ihre Verständlich- und Tauglichkeit geprüft wurden. Das Ergebnis waren vier Leitfragen mit jeweils 4-6 Fragen für die *Check*-Liste²:

1. „Zu Beginn des Interviews möchte ich Sie bitten, mir zu beschreiben, was Sie mit Informatik verbinden.“
2. „Welche Aufgaben hätte Ihrer Meinung nach Informatikunterricht in der Grundschule?“
3. „Wo taucht Ihrer Ansicht nach Informatik im Leben von Grundschulkindern ganz konkret auf?“

Vor der vierten Frage wurde den Lehrpersonen eine Übersicht über die Inhalts- und Prozessbereiche der Empfehlungen für Bildungsstandards Informatik für die Sekundarstufe I der GI vorgelegt [GI08]:

4. „Sie finden hier einige Fachbegriffe zur Informatik. Welche Begriffe sind Ihrer Ansicht nach bedeutsam für die Grundschule und welche unterrichtlichen Inhalte verbinden Sie mit diesen?“

Die Anordnung der Fachbegriffe (Bereiche) wechselte zwischen den Interviews, sodass

² Aus Platzgründen werden hier nur die Leitfragen aufgeführt.

gleiche Antworten nicht auf dieselbe Anordnung zurückgeführt werden können. Die Interviews wurden mit schriftlichem Einverständnis der Interviewten digital-auditiv aufgezeichnet und anhand von Anonymisierungs- und Transkriptionsregeln transkribiert.

3 Zwischenergebnisse und Ausblick

Bei den interviewten Lehrpersonen, nennen wir sie ALICE und BOB, existieren erste Präkonzepte zur Informatik und zu Informatikunterricht. Dies entspricht den Ergebnissen der Pilotstudie von Best und Marggraf [BM15] sowie der Studie von Funke, Geldreich und Hubwieser [FGH16].

ALICE verfügt in den Bereichen Ort der informatischen Bildung und Aufgaben von Informatikunterricht bereits über sehr konkrete Vorstellungen, die zum einen eng an den Mathematikunterricht geknüpft und zum anderen durch die hohe Durchdringung der Schülerschaft mit Smartphones geprägt sind. Die Aussagen von BOB hingegen scheinen noch nicht derart gefestigt zu sein. Er sieht den Ort der informatischen Bildung einerseits in einem fächerintegrierten Ansatz und andererseits in einem eigenständigen Fach Informatik („oder AG/Neigungsgruppe“). Die Frage, welche informatischen Inhalte wo behandelt werden sollten, führt er auf deren kognitive Komplexität und die intrinsische Motivation der Schülerinnen und Schüler³ zurück. Die Abgrenzung scheint jedoch nicht trennscharf. Sowohl ALICE als auch BOB verorten den Kontakt von Grundschulkindern mit Informatik in deren Interaktion mit Informatiksystemen, insbesondere Smartphones. In Punkto der Ausprägung dieses Kontakts scheinen sich die Schülerschaften der beiden Lehrpersonen zu unterscheiden. ALICE gibt an, dass die SuS mehrheitlich mit diesen Systemen in Kontakt treten. BOB hingegen beobachtet dies nur bei wenigen SuS. Dennoch sehen beide diesen Umstand durchaus kritisch. Aufgabe eines Informatikunterrichts in der Grundschule sei das Aufzeigen der Funktionsweise, aber auch der potentiellen Gefahren dieser Informatiksysteme.

Auch bezüglich der Inhalts- und Prozessbereiche [GI08] weichen die Aussagen von ALICE und BOB voneinander ab. Während ALICE erneut den starken Bezug zum Lehrplan bzw. dem Modellierungskreislauf der Mathematik herstellt sowie die Algorithmik hervorhebt, erarbeitet sich BOB ein Verständnis der Begriffe *ad-hoc*. Einige Assoziationen, wie die Verknüpfung des Bereichs Sprachen und Automaten mit „Software zum Erlernen von Fremdsprachen oder der Deutschen Sprache“, überraschen. Bob stellt jedoch auch Bezüge zu Programmiersprachen, Datenschutz sowie zur Datenspeicherung und -sicherheit her.

Die ersten beiden ausgewerteten Interviews lassen vermuten, dass die Vorstellungen der Lehrpersonen zur Informatik und zu Informatikunterricht von unterschiedlichen Faktoren geprägt werden. Dazu könnten Erfahrungen mit dem Informatikunterricht in der eigenen Schulzeit, Kontakte mit universitären „Computerkursen“, Einsatz von

³ Im Folgenden mit SuS abgekürzt.

Informatiksystemen in bestehenden Grundschulfächern, Kontakte der eigenen Kinder mit Informatiksystemen sowie weitere Faktoren zählen. Auch zur inhaltlichen Erschließung der Fachbegriffe (Bereiche) werden unterschiedliche Zugänge und Strategien gewählt, welche sich aus Alltagsverständnissen oder bestehenden Grundschulfächern ergeben könnten.

Über die Auswertung weiterer Interviews und die systematische Entwicklung eines Kategoriensystems sollen im Verlauf des Projekts IGS weitere Erkenntnisse gewonnen werden. Es wird auf Techniken der Qualitativen Inhaltsanalyse nach Mayring sowie der *Grounded Theory* nach Glaser und Strauss zugegriffen werden.

Literaturverzeichnis

- [ACM12] Computer Science Teachers Association (CSTA) und Association for Computing Machinery (ACM): Computer Science K-8. Building a Strong Foundation. ACM Press, New York, S. V, 2012.
- [Be15] Bender, E. et al.: Identifying and formulating teachers' beliefs and motivational orientations for computer science teacher education. *Studies in Higher Education* Vol. 41/2016 Issue 11, S. 1958-1973, 2015.
- [BDM10] Borowski, C.; Diethelm, I.; Mesaroş, A.: Informatische Bildung im Sachunterricht der Grundschule. Theoretische Überlegungen zur Begründung. *widerstreit-sachunterricht* Ausgabe 15/Oktober 2010, 2010.
- [BM15] Best, A.; Marggraf, S.: Das Bild der Informatik von Sachunterrichtslehrern. Erste Ergebnisse einer Umfrage an Grundschulen im Regierungsbezirk Münster. In (Gallenbacher, J. Hrsg.): *Informatik allgemeinbildend begreifen. INFOS 2015*. 16. GI-Fachtagung Informatik und Schule, Darmstadt 2015. Köllen Verlag + Druck, Bonn, S. 53-62, 2015.
- [DKW11] Diethelm, I.; Koubek, J.; Witten, H.: IniK – Informatik im Kontext. Entwicklungen, Merkmale und Perspektiven. *LOG IN* 169/170, S. 100 f., 2011.
- [FGH16] Funke, A.; Geldreich, K.; Hubwieser, P.: Primary school teachers' opinions about early computer science education. In (Sheard, J.; Montero, C. S. Hrsg.): *Koli Calling 2016*. Proc. 16th Koli Calling Int. Conf. on Computing Education Research, Koli 2016. ACM Press, New York, S. 135-139, 2016.
- [GI08] Gesellschaft für Informatik e.V.: Grundsätze und Standards für die Informatik in der Schule. *Bildungsstandards Informatik für die Sekundarstufe I*. Beilage zu *LOG IN* 150/151, S. 11, 2008.
- [He11] Helfferich, C.: *Die Qualität Qualitativer Daten. Manual für die Durchführung qualitativer Interviews*, 4. Auflage, VS Verlag für Sozialwissenschaften, Wiesbaden, S. 182-189, 2011.

Wir bedanken uns bei allen Lehrerinnen und Lehrern, die sich an der Umfrage und den Interviews beteiligt haben. Unser Dank gilt weiterhin den Schülern und der Bezirksregierung Münster für ihre Unterstützung.

Aufbau des Internets: Vorstellungsbilder angehender Lehrkräfte

Andreas Dengel¹ und Ute Heuer²

Abstract: Heranwachsende haben ein natürliches Fragebedürfnis in Bezug auf ihre Umwelt. In der digitalen Informationsgesellschaft des 21. Jahrhunderts ist das Internet aufgrund der allgegenwärtigen Vernetzung von Menschen, Computern und Informationen ein omnipräsenter Bestandteil der kindlichen Umwelt. Hierdurch entstehende Fragen richten Kinder vorwiegend an eine Bezugsperson mit Lehrfunktion, welche im Grundschulalter zumeist durch die entsprechende Lehrkraft verkörpert wird. Um auf solche Fragen adäquat eingehen zu können, sind entsprechende Fachkompetenzen der Lehrkraft notwendig. Anhand einer kognitionspsychologischen Fundierung und unter Berücksichtigung des tatsächlichen physisch-topologischen Aufbaus des Internets wurde eine Studie zu entsprechenden Vorstellungsbildern entwickelt und durchgeführt. Die Ergebnisse der Untersuchung mit 167 Lehramtsstudierenden aller Schularten zeigen, dass die Vorstellungsbilder von angehenden Lehrkräften zum Aufbau des Internets oftmals auf drastischen Fehlannahmen basieren, welche bei Studierenden des Grundschullehramts besonders stark ausgeprägt sind. Die Diskussion der Untersuchungsergebnisse mündet in einem Forschungsdesiderat zur Begegnung dieses Fachkompetenzdefizits, insbesondere im Grundschulbereich.

Keywords: Informatik in der Grundschule, Aufbau des Internets, Vorstellungsbilder, Lehramtsstudium, Fehlvorstellung

1 Einführung

Nahezu alle aktuellen Bildungsinitiativen fordern die weiterführende Digitalisierung der Schulen. Das Strategiepaper „Bildung in der digitalen Welt“ sowie das Modellprojekt „Digitale Schule 2020“ erstreben die Weiterentwicklung und kompetente Nutzung der digitalen Möglichkeiten in Schulen in Form vernetzter technischer Hilfsmittel [Km16], [DS17].

Zu den allgemeinbildenden Inhalten der heutigen Informationsgesellschaft zählt gemäß dem Strategiepaper „Bildung in der digitalen Welt“ der Kultusministerkonferenz unter anderem auch, die Herausforderungen der „digitalen Revolution“ bewältigen zu können, was eine Weiterentwicklung des Unterrichts und eine entsprechende Qualifikation der Lehrkräfte erfordert [Km16]. Eine damit einhergehende Professionalisierung angehender Lehrkräfte ist unabdinglich, um eine konsistente Vermittlung von Lerninhalten in Bezug auf die digitale Welt zu gewährleisten, besonders, da Informatik in Deutschland längst nicht in allen Schularten als Pflichtfach integriert ist.

¹ Universität Passau, Lehr- und Forschungseinheit "Lehramtsausbildung Mathematik und Informatik", Innstr. 33, 94032 Passau, andreas.dengel@uni-passau.de

² Universität Passau, Lehr- und Forschungseinheit "Lehramtsausbildung Mathematik und Informatik", Innstr. 33, 94032 Passau, heuer@fim.uni-passau.de

In Deutschland lässt besonders die Grundschule bislang eine Implementierung der Informatik in den Lehrplan missen, was bei Grundschülerinnen und -schülern zu einem Informationsdefizit bezüglich bestehender Fragen führt: In einer Untersuchung von [BDW16] wurden 2594 Fragen von 600 Schülerinnen und Schüler der Grundschule gesammelt, davon allein 656 Fragen zum Thema Internet. Beispielhaft werden Fragen wie „*Can I rent or buy the Internet?*“, „*Is there a central computer for the Internet?*“, „*What is the source of the Internet?*“ und „*Can the Internet get broken?*“ genannt.

Allerdings sind besonders solche physisch nicht allgegenwärtigen Konzepte unserer digitalen Welt, wie der Aufbau des Internets, bei Schülerinnen und Schülern zumeist von Fehlvorstellungen geprägt (u.a. [DZ10]; [Pa05]; [He15]). So stellten sich in der qualitativen Untersuchung nach [DZ10] 40% der 23 befragten Schüler einen zentralen Rechner vor, über den das Internet läuft, Probanden der Studie von [Pa05] mit 340 griechischen Schülerinnen und Schülern zeichneten neben ähnlichen zentralisierten bzw. hierarchischen Strukturen (56%) oder Formen der Direktvernetzung (6%) auch Modelle mit komplett fehlender Kommunikation (29%).

Falsche Vorstellungsbilder finden sich aber nicht nur auf Seite der Lernenden: auch bei angehenden Lehrerinnen und Lehrern haben sich oft Fehlvorstellungen eingepreßt, wie eine Voruntersuchung zeigen konnte: Es wurden 30 Studierende verschiedener Lehramtsfächer (außer Informatik) und Schularten nach ihrer Vorstellung zum E-Mail-Versand befragt. Hierbei ließen sich die Angaben von 93,3% der Probanden einem naiven metaphorischen Modell des Internets nach [He15] zuordnen, beispielsweise als direkte Datenleitung zwischen zwei Computern oder dem Internet als „Wolke“/Blackbox, bzw. konnten als eine Topologie der lokalen Vernetzung (Stern, Bus, Hierarchie) identifiziert werden.

2 Entstehung von Vorstellungsbildern

Das Alter, in welchem Kinder das erste Mal in Kontakt mit dem Internet kommen, lässt sich durch dessen Allgegenwärtigkeit und unausweichliche Präsenz in allen Lebensbereichen nicht klar bestimmen. Dadurch entwickeln Kinder teilweise schon sehr früh ihre eigene Vorstellung zum Aufbau des Internets. Hieraus ergibt sich zunächst die Frage, wie ein Vorstellungsbild zu einer bestimmten Thematik ohne konkrete Wissensvermittlung durch einen Lehrenden entsteht. Die Lernpsychologie nennt hierzu den von Gagné geprägten Begriff des kumulativen Lernens, welcher den Prozess des Verknüpfens neu erworbener Informationen mit bereits im Gedächtnis existierendem Wissen bezeichnet [Br79].

Im Modell des kumulativen Lernens wird das bereits existierende Wissen als „Schema“ und der Prozess der Eingliederung und Vernetzung neuer Informationen als „Assimilation“ bezeichnet [Ma81].³ Kumulatives Lernen erfolgt nach Mayer, wenn von außen kommende, neue Informationen folgende Schritte durchlaufen:

³ Die Verwendung der Begriffe *Schema / schema* und *mentales Modell / mental model* ist im deutsch- und englischsprachigen Raum nicht einheitlich. Da dieses Paper Problemlöseprozesse fokussiert (im Gegensatz zu subjektiven Funktionsmodellen anderer Untersuchungen), wird der Begriff *Schema* hier im Sinne von bereits existierendem Wissen nach [Ma81] verwendet.

- (1) *Aufnahme.* Zunächst muss der Lernende seine Aufmerksamkeit auf die eintreffende Information richten, sodass diese vom sensorischen ins Kurzzeitgedächtnis gelangt.
- (2) *Verfügbarkeit.* Der Lernende muss über zur neuen Information passende bereits vorhandene Konzepte im Langzeitgedächtnis verfügen, um die neue Information assimilieren zu können.
- (3) *Aktivierung.* Der Lernende muss aktiv sein bereits bestehendes Vorwissen nutzen, um die neue Information in sein Schema im Langzeitgedächtnis einzugliedern.

Ein besonderes Augenmerk sei hier auf den in (2) beschriebenen Prozess des Suchens nach einem passendem Schema gelegt: Diese „appropriate anchoring ideas“ werden in verschiedenen Kontexten von Lehrenden benutzt, um bei der Einführung von neuem Stoff an die Vorstellungen der Lernenden anzuknüpfen (u.a. [Ma81]; [He15]; [Fr01]). Gerade jedoch bei der selbstgesteuerten Aneignung neuer Inhalte kommt es vor, dass vermeintlich zur Information passende Schemata nicht der realen Struktur entsprechen, wodurch ein neues, fehlerhaftes Schema entsteht. Derart inkonsistente Schemata behindern durch ihre Funktion als „skelettartige Wissensstrukturen, die mit den Spezifika einer aktuellen Problemstellung angereichert werden, wenn die Person einem passenden Problemtyp begegnet“, nach [WM09] nicht nur den effektiven Problemlöseprozess, sondern liefern bei einem Mismatch zwischen Realität und Vorstellung auch ein falsches Ergebnis (beispielsweise ein Nichtmuttersprachler, der versucht, die Grammatikstrukturen seiner Muttersprache auf die deutsche Sprache anzuwenden).

Hier zeigt sich deutlich, dass bei Inhalten, deren selbstgesteuerte Aneignung intuitiv zu falschen Ergebnissen führen würde, eine Anleitung des Lernenden durch einen Experten zwingend erforderlich ist. Erfolgt eine Wissensvermittlung aufgrund einer fehlenden Lehrperson oder deren fehlender Fachkompetenz nicht, so bildet sich durch Assimilation anhand eines selbstständigen kumulativen Lernprozesses ein falsches, wenngleich äußerst persistentes Schema. Dieses wird für alle folgenden Lernprozesse der entsprechenden Thematik als Grundlage genommen. Ein möglicher Teilaspekt für die Konstruktion eines adäquaten Schemas der topologischen Struktur des Internets wird nachfolgend dargestellt.

3 Struktur des Internets

[KR12] sprechen von einem Puzzle, welches es bezüglich des Entwurfs einer Struktur des Internets zu lösen gilt: In einem ersten Schritt sind eine Milliarde Endnutzer und Dienstleister mit Internet-Service-Providern (ISPs) zu vernetzen. Die Hauptarbeit besteht jedoch darin, die ISPs selbst untereinander zu vernetzen. „This is done by creating a network of networks - understanding this phrase is the key to understanding the Internet.“ Die Tier-1-ISPs, auch Internet-Backbones genannt, befinden sich auf der obersten Stufe der ISP-Hierarchie und sind grundsätzlich auf die gleiche Art und Weise aufgebaut und vernetzt wie jedes Netzwerk: Sie besitzen Leitungen und Router und sind mit anderen Netzen verbunden. Ein Tier-1-ISP agiert global, es kann allerdings im Sinne einer vertikalen Integration gleichzeitig auch als ISP einer niedrigeren Stufe fungieren und noch niedrigstufigeren (z.B. regionalen) ISPs direkt Internetzugänge verkaufen [KR12]. In gewisser Weise „parallel zu den Tier-1-ISPs“ und „parallel zum Internet“ betreiben große Content Provider

(wie z.B. Google) ihr eigenes privates, weltumspannendes Netz von Daten-Zentren. Ein Nicht-Tier-1-ISP muss, um einen Teil des weltweiten Internets zu erreichen, zwar immer wieder seinen Datenverkehr durch einen der Tier-1-ISPs führen, versucht sich aber oft aus wirtschaftlichen Gründen auch direkt mit anderen ISPs gleicher (oder niedrigerer) Stufe zu verbinden. In diesem Fall können Daten übertragen werden, ohne ein Tier-1-Netz passieren (und dieses bezahlen) zu müssen [KR12]. Für einen schnellen, kostengünstigen und zuverlässigen Datenaustausch zwischen ISPs sorgen sogenannte Peering-Points, wie der in Frankfurt beheimatete DE-CIX, an welchen nach eigenen Angaben heute mehr als 550 Netzbetreiber (ISPs und Content Provider Netze) aus mehr als 50 Ländern angeschlossen sind [Br15]. Somit lässt sich mit [KR12] sagen, dass das heutige Internet - ein Netz aus Netzen - eine komplexe Struktur besitzt. Die nachfolgend beschriebene Untersuchung fokussiert in Anlehnung an [He15] nur folgenden Aspekt der Struktur: „Es gibt Netze, die verschiedenen Unternehmen gehören. Diese Unternehmen betreiben eigene Vermittlungspunkte und vernetzen diese, um Anfragen und Antworten innerhalb ihres Bereichs weiterzuleiten. Die Unternehmensnetze wiederum tauschen an gemeinsamen Punkten Daten aus. So passiert eine Anfrage typischerweise einige Unternehmensnetze bis sie schließlich an ihr Ziel gelangt.“

Ausgehend von den Vorstellungsbildern aus der Voruntersuchung, den Ergebnissen der Kognitionspsychologie und dem beschriebenen Aspekt der Struktur des Internets ergibt sich ein Forschungsdesiderat mit der Frage nach der Relevanz dieser Fehlvorstellungen bei angehenden Lehrkräften allgemein, sowie insbesondere in Bezug auf die Primarstufe, um dort aufkommenden Fragen der Heranwachsenden begegnen zu können. Da bereits frühere Untersuchungen das Vorhandensein von Fehlvorstellungen bezüglich der Struktur des Internets bei Erwachsenen im Allgemeinen nachweisen konnten (u.a. [TG98]), konzentriert sich die vorliegende Studie ausschließlich auf angehende Lehrkräfte. Anhand der dargestellten Struktur des Internets in Verbindung mit möglichen falsch erstellten Schemata aus der Lebenswelt der Studierenden und anhand der bereits erhaltenen Ergebnisse der Voruntersuchung wurden spezifische Fehlvorstellungen selektiert.

4 Erhebung der Vorstellungsbilder

4.1 Aufbau des Fragebogens

Der Fragebogen wurde in drei Teile untergliedert. Nach der Erhebung allgemeiner Informationen (Geschlecht, Schulart und Hauptfach/vertieftes Fach) wurden im zweiten Teil die eigenen Vorstellungsbilder der Studierenden bezüglich des Internets anhand einer Freitextfrage zum Aufruf einer Website und einer hierzu zu ergänzenden schematischen Zeichnung abgefragt. Zunächst sollte die Aufgabe *Sie rufen zuhause von Ihrem Computer A die Website der Uni Passau vom Server B auf. Beschreiben Sie zunächst kurz in Worten, welchen Weg entlang der Kabelverbindungen Ihre Anfrage bzw. die Antwort im Internet wählt.* bearbeitet werden.

Anschließend sollte die der Beantwortung zugrundeliegende Vorstellung mithilfe einer zu ergänzenden Zeichnung illustriert werden: *Zeichnen Sie folgend ein Schaubild, welches*

die Vernetzung von Computern/Servern (Kreise) im Internet darstellt (Kabelverbindungen). Dieses sollte zu Ihrer Beschreibung in 1. passen. Falls Sie neue Symbole einführen, beschriften Sie diese. Bei der vorgegebenen Zeichnung waren der eigene Computer und der Server der Universität als mit A bzw. B beschriftete Kreise eingezeichnet. Zur besseren Orientierung und zur Verdeutlichung der physischen Struktur waren noch weitere Kreise (gemäß der Aufgabenstellung Computer/Server) ohne Beschriftung eingezeichnet. Der dritte Teil bestand aus einer Selbstzuordnung, bei der die Probanden ihre selbst angegebene Vorstellung im Single-Choice-Verfahren einer schematischen Darstellung zuordnen sollten: Wählen Sie nun bitte die Darstellung aus, die Ihrer Vorstellung in 1. und 2. am Nächsten kommt. Bitte versuchen Sie nicht, eine richtige Struktur zu „erraten“, sondern beurteilen Sie ausschließlich Ihre in 1. und 2. angegebene Struktur. Die Darstellungen orientierten sich an den von Hennecke aufgeführten naiven metaphorischen Modellen / Modellen der Lokalvernetzung und den von Diethelm und Zumbrägel ermittelten Schülervorstellungen ([He15]; [DZ10]). Zur Veranschaulichung wurde das jeweilige Modell ohne namentliche Nennung grafisch und textuell dargestellt:

1. DV: *Direktverbindung / vollvermaschtes Netz*
2. Stern: *zentraler Computer / Sterntopologie*
3. Hierarchie: *hierarchischer Aufbau / erweiterter Stern*
4. Bus: *Bustopologie*
5. Router: *Netz von Routern*
6. Autonome Systeme: *Netz von Autonomen Systemen / Netz von Routern mit Peering*

Die Zuordnung des Modells sollte nur auf die im vorangegangenen Teil angegebenen Vorstellung bezogen werden, die Fragestellung wies hierauf ausdrücklich hin. Diese angegebene Methode wurde gewählt, um zum einen eine Zuordnung durch einen Korrektor zu vermeiden und zum anderen, um den Probanden die Möglichkeit zu gewähren, nicht in der eigenen Darstellung enthaltene Vorstellungen zu ergänzen. Um die Möglichkeit einer Fehlinterpretation der Fragestellungen trotz entsprechenden Wissens zu überprüfen, wurde eine Vergleichsstichprobe ausschließlich mit Lehramtsstudierenden mit dem Fach Informatik gebildet, bei welchen davon ausgegangen werden konnte, dass das notwendige Wissen zur korrekten Beantwortung der Fragen bereits erworben wurde. Hierbei gaben 13 von 14 Probanden eine korrekte Antwort, große Abweichungen zwischen eigener Darstellung und angekreuzter Topologie konnten nicht festgestellt werden.

4.2 Stichprobe und Durchführung

Insgesamt nahmen 167 Lehramtsstudierende an der Befragung teil, davon waren 30 männlich und 137 weiblich. Bei der Angabe der Schulart war mit 126 Angaben die Grundschule am häufigsten vertreten, gefolgt vom Gymnasium mit 20 Studierenden, der Mittelschule mit 13 Studierenden und der Realschule mit 8 Studierenden. Von den 167 Fragebögen wurden 163 in die Auswertung miteinbezogen, auch wenn Antworten fehlten. Da für die Hauptuntersuchung lediglich Lehramtsstudierende ohne Informatik von Interesse waren,

wurden 4 Fragebögen nicht mit in die Auswertung einbezogen, da diese Probanden Informatik als vertieft studiertes Fach angaben.

Die Studierenden wurden unter Aufsicht von einem der Autoren in verschiedenen Seminaren und Vorlesungen befragt. Die Probanden wurden darauf hingewiesen, dass sie lediglich ihre eigene Vorstellung angeben und sich nicht etwa in der Gruppe beraten sollen. Die Bearbeitungszeit betrug acht Minuten. Die Selbstzuordnung wurde so an die Fragebogen geheftet, dass die dort angegebenen Modelle bei Bearbeitung des zweiten Teils noch nicht sichtbar waren, wodurch eine Suggestivwirkung vermieden werden sollte. Rückfragen zur Fragestellung waren erlaubt.

4.3 Auswertung

Von den 163 ausgewerteten Fragebögen ließen 9,6% die Felder der eigenen Vorstellung frei oder gaben an, keine eigene Vorstellung zu haben (danach trotzdem angekreuzte Strukturen wurden aufgrund der Suggestivwirkung nicht gewertet).

fehlend	DV	Stern	Hierarchie	Bus	Router	Autonome Systeme	Gesamt
16	32	57	9	5	26	18	163

Tab. 1: gewählte Topologien

Insgesamt 22,2% der Studierenden wählten bei der Selbsteinordnung eine Struktur der Direktvernetzung (19,2% Peer-to-Peer, 3,0% Bus), 39,5% entschieden sich für eine hierarchische Struktur (34,1% Stern und 5,4% erweiterter Stern/Hierarchie). Lediglich 26,4% der Studierenden ordneten ihre eigene Vorstellung einer Vernetzung von Routern zu (15,6% Netz von Routern, 10,8% Netz von Autonomen Systemen). Als richtige Vorstellung wurde das Bild einer dezentralisierten Vernetzung gesehen, repräsentiert durch die Antworten „Netz von Routern“ und „Netz von Autonomen Systemen“.

Insgesamt widersprachen nach Beurteilung zweier unabhängiger, qualifizierter Rater (Staatsexamensabschluss Informatik bzw. Bachelorabschluss Internet Computing) zwischen 15% (Rater A) und 14,5% (Rater B) der eigenen (gewerteten) Darstellungen der Probanden den letztlich angekreuzten Topologien. Es wurden lediglich sich völlig widersprechende und aus der jeweiligen Kategorie (Direktvernetzung, hierarchische Struktur, Vernetzung von Routern) fallende Angaben als Abweichung gezählt, wie die Angabe einer Direktvernetzung im Beschreibungsteil in Verbindung mit einer angekreuzten Sterntopologie. Als Interrater-Reliabilitätsmaß wurde ein Cohens-Kappa-Wert von $K = .668$ und somit eine beachtliche Übereinstimmung ermittelt. Im Folgenden werden im Zuge der Übersichtlichkeit nur die Werte für Rater A genannt, da zwischen beiden Ratings nur für die Auswertung unwesentliche Unterschiede bestehen.

Bei einer gewerteten Abweichung (insgesamt 24) wählten 62,5% der Probanden in der letzten Frage eine Topologie der dezentralisierten Vernetzung, was zu höchst signifikanten ($p = .000$ bei $df = 1$) Chi-Quadrat-Werten von $\chi = 14,799$ führt. Die Probanden ohne eigene Angabe zum Vorstellungsbild („fehlend“ in Tab. 1) wurden nicht mehr einbezogen.

	falsche Vorstellung	richtige Vorstellung	Gesamt
keine Abweichung	94	29	123
Abweichung	9	15	24
Gesamt	103	44	147

Tab. 2: Zusammenhang zwischen gewerteter Abweichung (Rater A) und gewählter Topologie

Entfernt man die 24 Probanden mit einem Widerspruch zwischen eigener Darstellung und angekreuzter Topologie sowie die 16 Probanden ohne eigene Angabe zum Vorstellungsbild, verbleiben 123 gewertete Fragebögen mit 29 richtigen Antworten.

DV	Stern	Hierarchie	Bus	Router	Autonome Systeme	Gesamt
29	54	6	5	22	7	123

Tab. 3: gewählte Topologien ohne Widersprüche zur eigenen Vorstellung (Rater A)

Teilt man die Stichprobe auf in „Lehramtsstudierende mit Schulart Grundschule“ und „Lehramtsstudierende anderer Schularten“, ergeben sich folgende Ergebnisse:

	falsche Vorstellung	richtige Vorstellung	Gesamt
anderes Lehramt	16	11	27
Grundschullehramt	78	18	96
Gesamt	94	29	123

Tab. 4: Zusammenhang zwischen widerspruchsfreier, richtiger Vorstellung und dem Studiengang Grundschullehramt (Rater A)

Hieraus ergibt sich eine signifikante Korrelation ($p = .017$ bei $df = 1$) zwischen den Eigenschaften „studiert Grundschullehramt“ und „hat eine falsche Vorstellung“ (keine Vorstellung einer Vernetzung von Routern) mit einem Chi-Quadrat Wert von $\chi = 5,656$. Der Vergleich fasst alle anderen Schularten zusammen.

Als Ergebnisse lassen sich festhalten:

- Lehramtsstudierende haben verschiedene Vorstellungen vom strukturell-topologischen Aufbau des Internets.
- Knapp ein Viertel der Lehramtsstudierenden hat ein valides eigenes Bild vom strukturell-topologischen Aufbau des Internets.
- Studierende des Grundschullehramts haben häufiger falsche Vorstellungsbilder bezüglich des strukturell-topologischen Aufbaus als Lehramtsstudierende anderer Schularten.

5 Diskussion und Ausblick

Ein Wissensdefizit bezüglich des topologischen Aufbaus des Internets ist in allen Schularten unverkennbar. Dass diese Informationslücken noch einmal verstärkt bei Studierenden

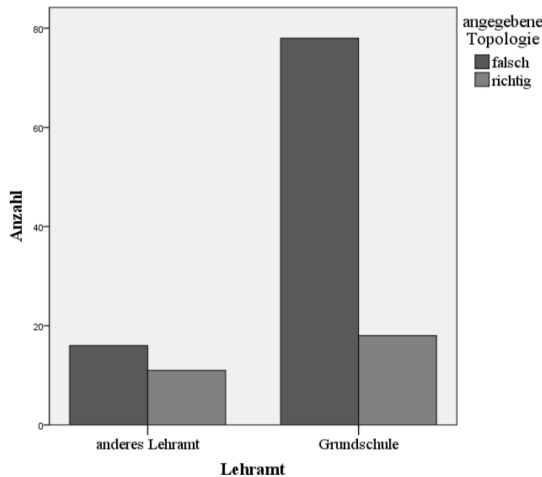


Abb. 1: Zusammenhang zwischen dem Studium Grundschullehramt / Vorstellungsbild

des Grundschullehramts hervortreten, ist besonders kritisch zu sehen: Zum einen entstehen gerade im Grundschulalter besonders viele Fragen zum Thema Computer und Internet, zum anderen gibt es in der Schulform der Grundschule bislang in Deutschland keinen Informatik-Unterricht. Diese Einflussfaktoren begünstigen das Entstehen eigener Schemata bei den Schülern, welche gemäß dem Prinzip des kumulativen Lernens als Grundlage für jedes weiter erworbene Wissen in diesem Themenbereich genommen werden.

Weiter ist jedoch erkennbar, dass bei einer direkten Wahl zwischen verschiedenen Möglichkeiten oft trotz eigener Fehlvorstellungen die richtige Option selektiert wird (das Treffen der richtigen Wahl bei Abweichung von der selbst angegebenen Struktur). Dies legt die Vermutung nahe, dass später erworbenes Wissen bezüglich korrekter Strukturen zwar aufgenommen, allerdings nicht in das eigene Vorstellungsbild übernommen wird, da eine Assimilation bei sich widersprechenden Vorstellungsbildern nicht möglich ist und diese Fehlvorstellungen somit persistent sind. Hieraus wird umso deutlicher, wie wichtig ein entsprechendes Entgegenwirken bereits in frühen Jahren ist, was zur Folge eine notwendige Stärkung der IT-Inhalte in der Primarstufe hat.

Die Angemessenheit der Platzierung des Fachs Informatik in der Grundschule konstatiert [Sc01], deren Notwendigkeit betonen u.a. [RR11] und [BDM10]. Zwar finden sich bereits im aktuellen Lehrplan Möglichkeiten im Rahmen des Sachkunde-Gegenstandsbereiches „Technik und Kultur“, diese werden in der Praxis jedoch aufgrund von fehlendem Selbstvertrauen / fehlender Fachkompetenz kaum wahrgenommen. Dem versucht beispielsweise das aktuelle Kooperationsprojekt „Informatik an Grundschulen“ der RWTH Aachen und der Universitäten Wuppertal und Paderborn entgegenzuwirken. Der Ausgangspunkt des Projekts folgt dem oben erörterten Ansatz: Es greift typische Alltagsfragen der Kinder zum Thema Informatik auf und entwickelt ausgehend von den individuellen Vorerfahrungen der Schülerinnen und Schüler Inhaltsblöcke zu den Themen „Umgang mit Informatio-

nen, Übertragung von Daten“, „Kryptologie, Einblicke in Konzepte der Verschlüsselung und Entschlüsselung“ und „Erste Schritte zur Programmierung“ [Ac15]. Grundlagen zum Aufbau und der Funktionsweise des Internets werden hier allerdings nur am Rande der Themen „Datenübertragung“ und „Binärdarstellung“ angeschnitten. Weitere Ansätze für handlungsorientierte Zugänge zur Informatik in der Grundschule bietet das Projekt „InfoSphere“ der RWTH Aachen mit auf die Grundschule zugeschnittenen Modulen wie „Zauberschule Informatik“, „Alles Informatik, oder was?!?“ und „Kryptografie zum Anfassen“ [SI17]. Der Fokus der beiden genannten Projekte liegt in der Förderung der informatischen Bildung von Grundschülerinnen und -schülern.

Das vorliegende Paper untersucht das Vorhandensein informatischer Grundlagen bei den hierfür intentional zuständigen Lehrkräften: Anhand der erhaltenen Untersuchungsergebnisse aus einem einzelnen, exemplarisch ausgewählten Teilbereich der Informatik konnte ein Hinweis auf ein spezielles Wissensdefizit bei angehenden Grundschullehrkräften gegeben werden. Der Erwerb eines Fachkompetenzkanons auf dem Gebiet der Informatik, sowohl korrespondierend zum untersuchten Inhalt als auch zu weiteren geeigneten Inhalten, ist allerdings aufgrund existierender Fragen seitens der Schülerinnen und Schüler indispensable. Um diesen Fragen zu begegnen und um die spätere Wissensentwicklung behindernde Fehlvorstellungen zu vermeiden, ist der Anteil informatischer Bildung im Primar-schulbereich zu stärken. Eine dauerhafte Forcierung der Informatik-Inhalte in der Grundschule würde selbstverständlich auch eine Implementierung von Inhalten der Informatik in die Curricula der Studiengänge des Grundschullehramts implizieren. Als Werkzeug zur Auswahl geeigneter Inhalte für die Modulkataloge des Grundschullehramts könnte das Konzept der fundamentalen Ideen nach [Br60] dienen (in Bezug auf die Informatik siehe auch [Sc93] bzw. [SS11]). Zur Strukturierung, Organisation und Präsentation dieser Inhalte müssen ausgehend von den Ansätzen der didaktischen Rekonstruktion (Model of Educational Reconstruction) nach [Du12] auch die Fehlvorstellungen der Lernenden, hier der angehenden Lehrkräfte der Grundschule, berücksichtigt werden. Nur durch Konzeption und Integration eines entsprechenden Modulkatalogs können die notwendige Fachkompetenz der angehenden Grundschullehrkräfte entwickelt, das Aufkommen von Fehlvorstellungen bei Kindern vermieden und entstehende Fragen adäquat beantwortet werden.

Literaturverzeichnis

- [Ac15] Acht, R.: Wie passt ein Video durchs Kabel? Informatische Bildung im Primarbereich. Schule NRW 07/08/15, S. 327–329, 2015.
- [BDM10] Borowski, C., Diethelm, I., Mesaros, A.: Informatische Bildung im Sachunterricht der Grundschule: Theoretische Überlegungen zur Begründung, <http://www.widerstreit-sachunterricht.de/ebene1/superworte/infor/BorDieMe.pdf>, Stand: 1.2.2017.
- [BDW16] Borowski, C., Diethelm, I., Wilken, H.: What children ask about computers, the Internet, robots, mobiles, games etc. In: Proceedings of the 11th Workshop in Primary and Secondary Computing Education, WiPSCE 2016, Münster, Germany, October 13-15, 2016. S. 72–75, 2016.
- [Br60] Bruner, J. S.: The Process of Education. Harvard University Press, 1960.

- [Br79] Bransford, J.: Human cognition: Learning, understanding, and remembering. Belmont, CA: Wadsworth, 1979.
- [Br15] Brühl, V.: Wirtschaft des 21. Jahrhunderts: Herausforderungen in der Hightech-Ökonomie. Springer Fachmedien, Wiesbaden, 2015.
- [Du12] Duit, R., Gropengießer, H., Kattman, U., Komorek, M., Parchmann, I.: The Model of Educational Reconstruction - a Framework for Improving Teaching and Learning Science. In: Science Education Research and Practice in Europe: Retrospective and Prospective, Jorde, D., Dillon, J. (Hrsg.) S.13-37, 2012.
- [DS17] Bildungspakt Bayern. Projekt: Digitale Schule 2020, <http://bildungspakt-bayern.de/digitale-schule-2020/>, Stand:31.1.2017.
- [DZ10] Diethelm, I., Zumbärgel, S.: Wie funktioniert eigentlich das Internet? Empirische Untersuchung von Schülervorstellungen. In: Proc. Didaktik der Informatik. Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik. Köllen Verlag, Bonn, S. 33–44, 2010.
- [Fr01] Freiman, T.: Kumulatives Lernen im Biologieunterricht. In: Praxis der Naturwissenschaften - Biologie in der Schule Jahrgang 50 Nr. 7, 2001.
- [He15] Hennecke, M.: Modellvorstellungen zum Aufbau des Internets. In: Tagungsband zur INFOS 2015, Lecture Notes in Informatics (LNI), 2015.
- [Km16] Bildung in der digitalen Welt, Beschluss vom 8.12.2016, https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2016/Bildung_digitale_Welt.Webversion.pdf, Stand: 20.12.2016.
- [KR12] Kurose, J.F., Ross, K.W.: Computer Networking. Addison-Wesley, 2012.
- [Ma81] Mayer, R.E.: The Psychology of How Novices Learn Computer Programming. In: Studying the Novice Programmer. Lawrence Erlbaum Associates, 1981.
- [Pa05] Papastergiou, M.: Students Mental Models of the Internet and Their Didactical Exploitation in Informatics Education. Education and Information Technologies, 10 (4), S. 341–360, 2005.
- [RR11] Romeike, R., Reichert, D.: PicoCrickets als Zugang zur Informatik in der Grundschule, 2011.
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 25, S. 20-31, 1993.
- [Sc01] Schwill, A.: Ab wann kann man mit Kindern Informatik machen? Eine Studie über informatische Fähigkeiten von Kindern. In: Keil-Slawik, R.; Magenheim, J. (Hrsg.): Informatikunterricht und Medienbildung: INFOS 2001, 2001.
- [SI17] Schülerlabor Informatik - InfoSphere, Informatik entdecken in Modulen für alle Schulformen und Klassenstufen, <http://schuelerlabor.informatik.rwth-aachen.de/module/grundschule>, Stand: 1.2.2017.
- [SS11] Schubert, S., Schwill, A.: Didaktik der Informatik. Spektrum Akad. Verl., Berlin Heidelberg, 2011.
- [TG98] Thatcher, A., Greyling, M.: Mental models of the Internet. In: International Journal of Industrial Ergonomics 22, S. 299-305, 1998.
- [WM09] Wild, E., Möller, J.: Pädagogische Psychologie. Springer-Verlag, Berlin Heidelberg, 2009.

Vom Lehrplan zur LehrerInnenbildung - Erste Erfahrungen mit obligatorischer Informatikdidaktik für angehende Schweizer PrimarlehrerInnen

Beat Döbeli Honegger¹, Michael Hielscher²

Abstract: In den letzten Jahren ist Informatikunterricht in verschiedenen europäischen Ländern in die Lehrpläne aufgenommen worden und soll zunehmend bereits im Grundschulalter beginnen. Auch der im Jahr 2015 verabschiedete »Lehrplan 21« für die 21 deutschsprachigen Kantone der Schweiz sieht bereits ab der 5. Klasse der Primarschule ein eigenes Zeitgefäß für »Medien und Informatik« vor. Diese Entwicklung stellt die Lehrerinnen- und Lehrerbildung vor große Herausforderungen. Die pädagogische Hochschule Schwyz (PHSZ) hat als erste Lehrerbildungsstätte in der deutschsprachigen Schweiz den Studiengang Primarschule entsprechend angepasst und erteilt künftig eine Lehrbefähigung »Medien und Informatik«. Im Herbstsemester 2016 wurde eine obligatorische Informatik-Lehrveranstaltung für alle Studierenden des Studiengangs Primarstufe durchgeführt. Bei der Eingangsbefragung (n=85) bestätigte sich die Hypothese, dass angehende Primarlehrpersonen weder über eine Vorstellung des Themas Informatik noch über das notwendige Fachwissen verfügen. Der Beitrag schildert die Entstehungsgeschichte des Fachs »Medien und Informatik« im Lehrplan 21 in der Schweiz und diskutiert exemplarisch anhand der Lehrveranstaltung die Herausforderungen einer Informatikdidaktik für Primarlehrpersonen.

Keywords: Lehrplan, Informatikdidaktik, Grundschule, Ausbildung, Weiterbildung.

1 Informatik in der Grundschule

Unser Alltag wird immer stärker von digitalen Medien geprägt. Der Einfluss digitaler Medien auf alle Aspekte ist so groß, dass verschiedene Experten die Erfindung des Computers mit der Erfindung des Buchdrucks vergleichen und von einem digitalen Leitmedienwechsel sprechen [Dö16]. Dies hat sowohl den Ruf nach Informatischer Bildung in der Schule (z.B. [Ga13]) als auch das Verständnis für diese Forderung in den letzten Jahren gestärkt (PM14). In immer mehr Ländern sind deshalb Bemühungen im Gange, Informatik stärker in der Schule zu verankern. Gleichzeitig ist auch die Tendenz zu beobachten, dass bei entsprechenden Projekten das Einstiegsalter für Informatik sinkt. War Informatik noch bis vor wenigen Jahren meist auf der Sekundarstufe I und II angesiedelt, wird heute immer häufiger bereits in der Grundschule oder noch früher begonnen. Auch im deutschsprachigen Teil der Schweiz ist eine solche Entwicklung im Gang.

¹ Pädagogische Hochschule Schwyz, IMS, Zaystrasse 42, CH-6410 Goldau, beat.doebeli@phsz.ch

² Pädagogische Hochschule Schwyz, IMS, Zaystrasse 42, CH-6410 Goldau, michael.hielscher@phsz.ch

1.1 Der Lehrplan 21 in der Schweiz

Das schweizerische Bildungssystem ist sehr föderalistisch. So sind z.B. Lehrpläne bisher kantonal geregelt. Im Jahr 2006 haben die Bildungsdirektoren der deutsch- und mehrsprachigen Kantone (D-EDK) beschlossen, einen sprachregionalen Lehrplan für die gesamte obligatorische Schulzeit (Kindergarten bis Ende Sekundarstufe I) zu entwickeln. Dieser wurde von 2010 bis 2014 erarbeitet und danach zur Umsetzung an die 21 beteiligten Kantone übergeben. Da der Lehrplan 21 nur Empfehlungscharakter hat, entscheiden alle 21 Kantone einzeln, ob, wie und wann er umgesetzt werden soll. In zahlreichen Kantonen finden aufgrund heftiger Kritik konservativer Kreise Volksabstimmungen zur Einführung des Lehrplans 21 statt.

Die Zahl 21 steht nicht nur für die 21 am Lehrplan beteiligten Kantone, sondern soll auch ausdrücken, dass der Lehrplan für das 21. Jahrhundert gedacht ist. Neben der versuchten Harmonisierung der kantonalen Lehrpläne lassen sich beim Lehrplan 21 vier wesentliche Veränderungstendenzen ausmachen [DEDK2014]:

- Orientierung an drei Zyklen (Kindergarten bis 2.Klasse, 3.-6. und 7.-9. Klasse) statt der Definition von spezifischen Jahrgangsziele
- Kompetenzorientierte Formulierungen statt bisheriger Lernziele
- Definition von Fachbereichen statt einzelner Fächer
- stärkere Betonung überfachlicher Themen und Kompetenzen

1.2 Informatik im Lehrplan 21

In der ursprünglichen Planung von 2010 war der Themenbereich Digitalisierung nur am Rande unter dem Titel »ICT und Medien« als eines von vielen überfachlichen Themen ohne eigene Stundendotation vorgesehen [DEDK10]. Dementsprechend waren informatische Kompetenzen im ersten veröffentlichten Lehrplanentwurf eher versteckt und ohne die explizite Nennung des Begriffs Informatik vertreten [DEDK13]. Obwohl versteckt, führten diese Kompetenzbeschreibungen zu einer breiten bildungspolitischen Diskussion (siehe z.B. [Bu13]). Insbesondere das in der Öffentlichkeit wahrgenommene Programmieren im Schulunterricht wurde kontrovers diskutiert. Nachdem sich zahlreiche Kreise für eine stärkere Verbindlichkeit digitaler Kompetenzen ausgesprochen haben (z.B.[ICTS13]), wurde der Lehrplanteil in »Medien und Informatik« (M+I) umbenannt und zu einem Modul umgewandelt. Der bis zu diesem Zeitpunkt im Volksschulbereich undefinierte Begriff »Modul« stellt dabei einen bildungspolitischen Kompromiss dar, da sich ein neues Fach vermutlich nicht hätte realisieren lassen.

Der im Jahr 2015 als Teil des Lehrplans 21 von der D-EDK verabschiedete und zur Umsetzung den Kantonen übergebene Teillehrplan »Medien und Informatik« [DEDK15] unterscheidet folgende drei Kompetenzbereiche:

- **Anwendungskompetenzen:** „Schülerinnen und Schüler nutzen Informations- und Kommunikationstechnologien in allen Fach- und Lebensbereichen effektiv und effizient.“
- **Medien:** „Schülerinnen und Schüler können an der Mediengesellschaft selbstbestimmt, kreativ und mündig teilhaben und sich sachgerecht und sozial verantwortlich verhalten.“
- **Informatik:** „Schülerinnen und Schüler verstehen Grundkonzepte der automatisierten Informationsverarbeitung, nutzen sie zur Entwicklung von Lösungsstrategien in allen Lebensbereichen und zum Verständnis der Informationsgesellschaft.“

Mit dieser Dreiteilung ist der Teillehrplan »Medien und Informatik« nicht weit entfernt vom im Februar 2016 entwickelten Dagstuhl-Dreieck (s. Abb. 1). Während im Lehrplan 21 bereits belegte Fachbegriffe verwendet werden, werden im Dagstuhl-Dreieck historisch unbelastetere Perspektiven definiert.

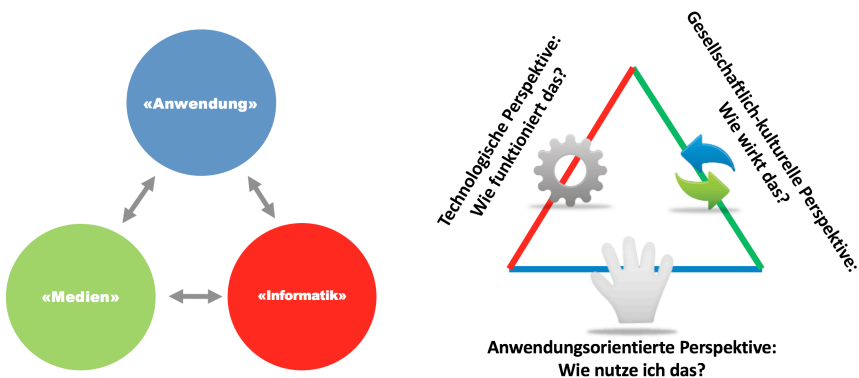


Abb. 1: Ähnliche Dreiteilung digitaler Kompetenzen im Teillehrplan »Medien und Informatik« [DEDK15] (links) sowie im Dagstuhl-Dreieck [Br16] (rechts)

Für zwei der drei Bereiche (Medien, Informatik) gibt der Lehrplan 21 einen Kompetenzaufbau vor. Die Kompetenzen im dritten Bereich »Anwendung« sollen hingegen in anderen Fachbereichen integriert unterrichtet werden und sind damit nicht unmittelbarer Gegenstand des eigenständigen Zeitgefäßes. Der Bereich Informatik gliedert sich wiederum in die drei Teilbereiche Datenstrukturen, Algorithmen und Informatiksysteme. Die darin aufgeführten Kompetenzen wurden teilweise jedoch aus bildungspolitischen Gründen gewählt und folgen damit weniger fachdidaktischen Überlegungen.

Aufgrund des Empfehlungscharakters des Lehrplans 21 ist es den Kantonen überlassen, wie sie den Begriff Modul bezüglich Stundenplan und Zeugniseintrag interpretieren. Ebenso ist offen, ob die Inhalte des Moduls von allen Klassenlehrpersonen oder von speziellen Fachlehrpersonen unterrichtet werden, oder ob es zukünftig eine eigenständige Lehrbefähigung für Medien und Informatik geben soll. Während einige

Kantone ein eigenes Zeitgefäß bereits ab der 5. Klasse vorsehen (z.B. Zürich und St. Gallen), planen andere ein solches erst ab der 7. Klasse und integrieren die Inhalte auf der Primarstufe in bestehende Fächer wie Mathematik und Deutsch (z.B. Luzern oder Schwyz). Da in den Schweizer Primarschulen grundsätzlich das Klassenlehrerprinzip gilt, sind praktisch alle Primarlehrpersonen mit dem Thema M+I konfrontiert, können ihm also nicht ausweichen (anders in der Sekundarstufe I, in der Lehrpersonen nur eine Auswahl von Fächern unterrichten). Dass nun auf der Primarschulstufe alle angehenden und praktizierenden Lehrpersonen im Bereich Informatik aus- bzw. weitergebildet werden, ist im deutschsprachigen Raum bisher einzigartig.

1.3 Herausforderungen bei der Aus- und Weiterbildung von Lehrpersonen

Mit dem Lehrplan 21 gewinnt das Thema Medien und Informatik stark an Bedeutung. Damit benötigen Lehrpersonen aber auch eine entsprechende Aus- oder Weiterbildung. Dies stellt die Kantone und die pädagogischen Hochschulen (wo in der Schweiz auch die Weiterbildung von Lehrpersonen angesiedelt ist) vor zahlreiche Herausforderungen:

1. **Fehlendes Fachwissen und Erfahrung bei Lehrpersonen:** Amtierende und angehende Lehrpersonen verfügen insbesondere im Bereich Informatik über praktisch kein Vorwissen und haben auch keine Unterrichtserfahrungen aus der eigenen Schulzeit [Döb15].
2. **Fehlendes Fachwissen und Erfahrung bei Dozierenden an Pädagogischen Hochschulen:** Derzeit sind Dozierende in der LehrerInnenbildung mit fundierter Informatikausbildung selten. Selbst diesen Dozierenden fehlt jedoch meist entsprechende Unterrichtserfahrung auf der Zielstufe.
3. **Fehlende Lehrmittel:** Derzeit sind entsprechende, auf den Modullehrplan »Medien und Informatik« abgestimmte Lehrmittel erst in Entwicklung und werden anfänglich für die Aus- und Weiterbildung nicht zur Verfügung stehen.
4. **Fehlende Zeit für Aus- und Weiterbildung:** Sowohl aus finanziellen Gründen als auch aufgrund der sonstigen Belastung von Lehrpersonen steht weniger Aus- und Weiterbildungszeit für »Medien und Informatik« zur Verfügung als es Fachdidaktiker als notwendig ansehen.
5. **Reservierte Haltung der Studierenden:** Lehramtsstudierende haben gegenüber dem Thema Medien und Informatik oft eine reservierte bis ablehnende Haltung (siehe auch ähnliche Ergebnisse aus deutschen Studien, z. B. [KB11]).
6. **Fehlendes Bewusstsein bei Entscheidungsträgern:** Sowohl politischen als auch administrativen Entscheidungsträgern fehlt anfänglich oft noch das Bewusstsein für das Thema und seine Aspekte. Insbesondere die Unterscheidung zwischen Informatik als eigenständige Disziplin und der allgemeinen Nutzung von Computern im Unterricht fällt häufig schwer.

2 Situation an der Pädagogischen Hochschule Schwyz

Die Pädagogische Hochschule Schwyz (PHSZ) ist eine kleine Pädagogische Hochschule mit vierfachem Leistungsauftrag (Ausbildung, Weiterbildung, Dienstleistung, sowie Forschung und Entwicklung) in Goldau (Kanton Schwyz) mit ca. 100 Mitarbeitenden. Sie bietet in der Ausbildung die beiden Studiengänge »Lehrperson für die Primarstufe« (1.-6. Klasse) und »Lehrperson für Kindergarten/Unterstufe« (Kindergarten bis 2. Klasse) und hat aktuell ca. 350 Studierende. Im April 2016 hat der Hochschulrat die beiden Studiengänge aufgrund des Lehrplans 21 angepasst. Dabei stand insbesondere die Erweiterung der Studiengänge um den Fachbereich M+I im Zentrum. Der Studiengang Primarschule enthält neu 6 ECTS-Punkte (von 180) in fünf Modulen im Bereich Medien und Informatik. Zusätzlich werden mediendidaktische Kompetenzen in einer weiteren Veranstaltung aufgebaut (Modulübersicht siehe Abb. 2).

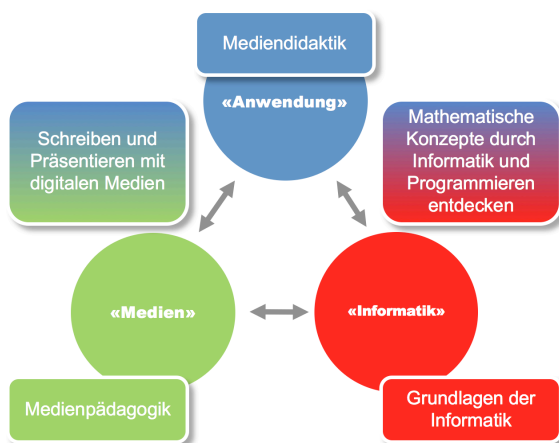


Abb. 2: Modulverteilung des Fachkerns »Medien und Informatik« an der PHSZ

Die PHSZ ist damit die erste Pädagogische Hochschule in der deutschsprachigen Schweiz, die im Herbstsemester 2016 mit der Ausbildung gemäß Lehrplan 21 im Bereich M+I begonnen hat. Insbesondere ist sie die erste Deutschschweizer PH mit einer obligatorischen Veranstaltung in Informatikdidaktik für alle Studierenden der Primarstufe. Vom Kanton Schwyz wurde die PHSZ mit der Weiterbildung der amtierenden Lehrpersonen für den Lehrplan 21 beauftragt. Für Primarlehrpersonen der 5. und 6. Klasse stellt dabei die obligatorische Weiterbildung in Informatik im Umfang von 10 Halbtagen den größten Themenbereich dar.

3 Erstmalige Durchführung der Lehrveranstaltung »Grundlagen der Informatik«

Im Herbstsemester 2016 haben die beiden Autoren erstmalig die obligatorische Lehrveranstaltung »Grundlagen der Informatik« im ersten Semester des Studiengangs Primarstufe durchgeführt. Vor Beginn der Lehrveranstaltungen haben zwei Befragungen stattgefunden, einerseits die allgemeine Eingangsbefragung der PHSZ [Eck16], andererseits eine lehrveranstaltungs-spezifische der beiden Autoren dieses Beitrags.

Von den 85 Studierenden der Lehrveranstaltung sind ca. 80% weiblich, 95% in der Schweiz geboren und über 85% sprachen zu Hause deutsch oder Schweizerdeutsch. 35% verfügten bei Studienbeginn bereits über eine abgeschlossene Berufsausbildung und beschränkten mit dem Beginn des Studiums somit den zweiten Bildungsweg [Eck16]. Bereits die allgemeine Eingangsbefragung gibt erste Hinweise auf die Motivation der Studierenden in Bezug auf digitale Medien. So rangiert der thematische Schwerpunkt der PHSZ im Bereich digitaler Medien auf Platz 7 von 9 möglichen Gründen für die Wahl des Studienorts. Zudem möchten sich mehr Studierende möglichst viel pädagogisches Hintergrundwissen aneignen und ihre Lern- und Arbeitsstrategien im Studium verbessern, als ihre Kompetenz im Bereich digitaler Medien verbessern (siehe Abb. 3). Diese Frage ist insofern relevant, als dass die PHSZ einen Schwerpunkt im Bereich digitaler Medien hat und zu den ersten Pädagogischen Hochschulen in der Schweiz gehörte, die bereits 2004 ein Notebookobligatorium für Studierende einführte.

Was trifft auf Sie am ehesten zu?

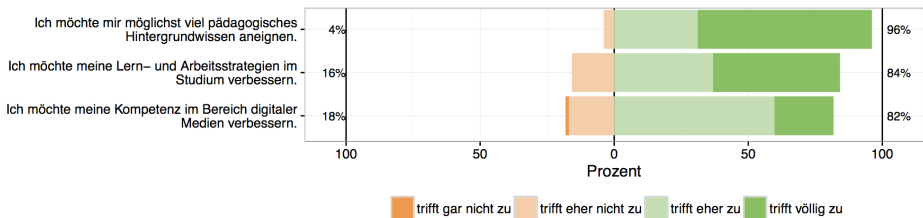


Abb. 3: Ziele für das Studium der Studierenden der Lehrveranstaltung »Grundlagen der Informatik« [Eck16]

Während in der Eingangsbefragung des Studiengangs noch sehr allgemein von »neuen Medien« und »digitalen Medien« die Rede war, interessierten in der Eingangsbefragung der Lehrveranstaltung mehr die Vorstellungen und Erwartungen der Studierenden zum Seminar und der Informatik in der Primarschule sowie ihre persönlichen Erfahrungen und Anwendungskompetenzen.

Mehr als ein Drittel der Studierenden gab an, dass sie auf ihren Notebooks noch keine oder wenige persönliche Daten abgelegt haben, womit davon ausgegangen werden kann, dass die Geräte erst mit Beginn des Studiums angeschafft wurden. Alle 85 Studierenden gaben an, ein Smartphone zu besitzen.

Die Antworten auf die Frage *“Was stellen Sie sich unter Informatik in der Primarstufe vor? Beschreiben Sie stichwortartig.”* wurden in Tabelle 1 thematisch gruppiert und nach Anzahl Nennungen sortiert. Bis auf das Programmieren beziehen sich alle Antworten auf Anwendungskompetenzen, die laut Lehrplan 21 nicht zur Informatik gehören und unabhängig davon in allen Fächer aufgebaut und angewendet werden sollen.

Thematisch gruppierte Freitextantworten	Nennungen
Umgang mit Office-Programmen erlernen (z.B. Textverarbeitung)	38
Lernprogramme / Lernspiele anwenden	32
Handhabung von Computer / ICT erlernen	30
Umgang mit dem Internet erlernen (inkl. Sicherheit, Datenschutz)	24
Tastaturschreiben erlernen (10-Finger-System)	20
Webrecherche erlernen (Computer als Informationsquelle nutzen)	12
Bild- und Videobearbeitung erlernen	5
Einsatz von ICT im Schulalltag thematisieren	4
Programmieren lernen	4

Tab. 1: Von erstsemestrigen PH-Studierenden erwartete Inhalte bei »Informatik in der Primarschule«, absteigend geordnet nach Anzahl Nennungen

Die Antworten zeigen, dass ein Verständnis der Informatik als eigenständige Disziplin und die damit verbundenen Fachinhalte beim überwiegenden Teil der Studierenden fehlt. Dies ist wenig erstaunlich, weisen doch selbst Untersuchungen zu Abbruchquoten in Informatikstudiengängen ebenfalls auf das Problem von Fehlvorstellungen hin, obwohl diese Studierenden ein offensichtliches Interesse am Thema haben (vgl. [BM05], [HSW06]).

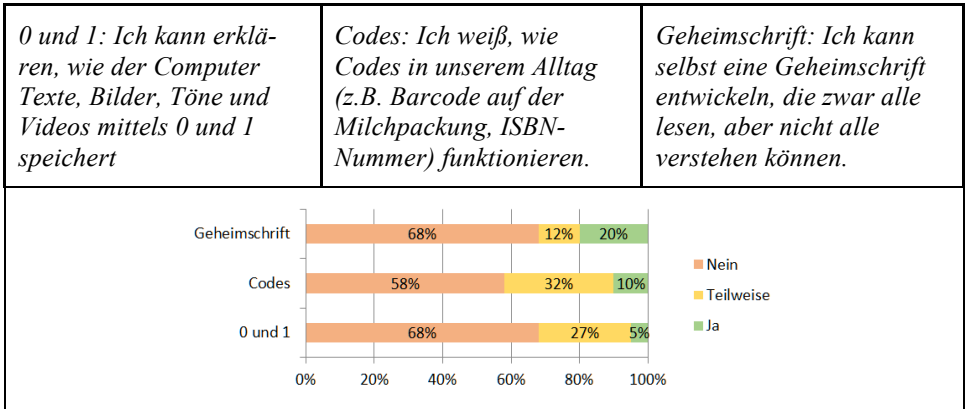
In der Eingangsbefragung wurde erhoben, ob die Studierenden schon einmal etwas selbst programmiert haben (29% ja, 71% nein). Bei den 25 Studierenden, die “ja” angaben, wurde in zwei Folgefragen mit Freitextantworten konkretisiert, was und in welcher Programmierumgebung programmiert worden ist. Tabelle 2 zeigt den Versuch einer thematischen Gruppierung der Antworten. Die Nennung verschiedenster Anwendungsprogramme deutet stark darauf hin, dass der Begriff “Programmieren” bei rund der Hälfte dieser Studierenden unklar ist und mit “Installieren” oder “Konfigurieren” gleichgesetzt wurde. Auch die Antworten “DropBox”, “USB-Stick”, “Schule” oder “Apple” auf die Frage der eingesetzten Programmierumgebung deuten auf fehlende oder falsche Vorstellungen im Bereich des Programmierens hin. Insgesamt gaben 6 der 25 Studierenden mit Programmiererfahrung an, bereits mit Scratch gearbeitet zu haben (gefolgt von Java und Python mit je 2 Nennungen). Insgesamt verfügten somit nur rund 10% aller Studierenden über Vorerfahrungen im Bereich des Programmierens.

Thematisch gruppierte Freitextantworten	Nennungen
Outlook, Adobe Reader, iTunes, Anti Virus, FortiClient ...	8
kleine Rechenprogramme programmiert (mathematische Themen)	7
Scratch	3
Kara	1
Datenbanken	1
Turtle	1
Simulation	1
Webseite erstellt	1
Vergleich von zwei Exceltabellen	1

Tab. 2: Programmiererfahrungen erstsemestriger PH-Studierender (Selbstauskunft)

Für drei ausgewählte, anspruchsvollere Kompetenzen aus dem Lehrplan 21 sollten sich die Studierenden einerseits selbst einschätzen und andererseits angeben, ob diese Kompetenzen für heutige Schülerinnen und Schüler Ende der 6. Klasse relevant seien. (siehe Tab. 3). Über 40% der angehenden Primarlehrpersonen waren der Meinung, dass Kinder erklären können sollten, wie der Strichcode auf einer Milchpackung funktioniert, dies aber nur rund 10% selbst erklären könnten. Auch im Bereich der Anwendungskompetenzen zeigt sich eine Diskrepanz zwischen dem Anspruch des Lehrplans 21 und den Selbsteinschätzungen der Erstsemestrigen. Bis auf wenige Ausnahmen konnten alle Studierenden das Betriebssystem ihres Notebooks und ihres Smartphones in einem Freitextfeld benennen. Eine Single-Choice-Frage zu Messgrößen der Informatik (*Wieviel Speicherplatz benötigen 3 Minuten Musik auf einem Smartphone?*) konnten hingegen nur noch rund 30% korrekt beantworten und mehr als die Hälfte der Studierenden wählten die Antwortoption "keine Ahnung". Zusammenfassend bestätigen die beiden Eingangsbefragungen somit die in Abschnitt 2.3 genannten Herausforderungen 1 und 5: Studierende verfügen über wenige Vorkenntnisse und sind nicht besonders interessiert am Thema. Insbesondere bezüglich Vorwissen unterscheidet sich damit das Thema Informatik von den meisten anderen Schulfächern.

Das einsemestrige Seminar gliederte sich in acht 90-minütige Präsenzveranstaltungen mit jeweils 20 bis 24 Teilnehmenden (in 4 Kursgruppen) und Selbststudienphasen. Inhaltlich orientierte sich die Veranstaltung eng am Lehrplan mit den drei Teilbereichen Datenstrukturen, Algorithmen und Informatiksystemen, wobei anteilig ein größeres Gewicht auf Algorithmen, insbesondere auf das Programmieren gelegt wurde. Zudem wurde in den ersten beiden Veranstaltungen anhand anschaulicher Unterrichtsbeispiele am Grundverständnis der Informatik als eigenständige Disziplin gearbeitet. In der letzten Präsenzveranstaltung wurden die Informatikdidaktik und ihre Besonderheiten im Rückblick auf das Semester thematisiert. Die detaillierten Lernziele, Kursplanung und Materialien sind im Veranstaltungswiki (phsz.doebe.li/GDI16) öffentlich einsehbar.



Tab. 3: Selbsteinschätzung angehender PrimarlehrerInnen bezüglich Informatik-Kompetenzen aus dem Lehrplan 21

Gegen Ende des Semesters wurde eine Kursevaluation durchgeführt. Die Lehrveranstaltung wurde insgesamt überwiegend als “gut” bis “sehr gut” bewertet. Die Seminarinhalte wurden als “eher anspruchsvoll” bis “sehr anspruchsvoll” bezeichnet. Die Studierenden sollten u.a. auch angeben, ob sie sich nun in der Lage sehen, eine typische Fragestellung/Problemstellung aus dem Themenbereich der Informatik zu bearbeiten. Dem konnte mehrheitlich nur bedingt zugestimmt werden. Insbesondere das Programmieren mit Scratch wurde von den Studierenden als zu schwierig und mit “zu wenig Zeit” beschrieben. Mehrheitlich wünschen sich die Studierenden deshalb mehr Präsenzveranstaltungen und eine intensivere Auseinandersetzung mit Scratch.

4 Dilemma von Motivation vs. Wissensvermittlung

Da die Inhalte des Modullehrplans M+I in einigen Schweizer Kantonen auf der Primarstufe integriert in den bestehenden Fächern unterrichtet werden sollen, besteht die bei integrierten Ansätzen übliche Gefahr von Vermeidungsstrategien bei Lehrpersonen, die diese Inhalte aus verschiedenen Gründen nicht unterrichten wollen oder können. Fehlt die extrinsische Vorgabe durch einen Fachlehrplan, hängt es an der intrinsischen Motivation der Lehrperson ab, ob sie informatische Themen unterrichtet. Da den Studierenden - wie oben gezeigt - eine Vorstellung des Themas Informatik fehlt, muss diese intrinsische Motivation in der Lehreraus- und -weiterbildung angelegt werden. Sowohl in der Aus- als auch in der Weiterbildung steht aber bereits zu wenig Zeit zur Verfügung, um die eigentlich notwendige fachliche und fachdidaktische Kompetenz zu vermitteln. Bei der Planung von Informatikaus- und -weiterbildungen entsteht somit das Dilemma, ob angesichts der begrenzten zeitlichen Ressourcen bei den angehenden und amtierenden Lehrpersonen eher die Motivation oder die informatischen Kompetenzen stärker gewichtet werden sollen (siehe Abb. 4). Eine zu starke Konzentration auf fach-

wissenschaftliche Inhalte der Informatik könnte dazu führen, das häufig bereits negativ vorgeprägte Bild der Informatik bei Lehrpersonen zu verstärken und damit intrinsische Motivation zu verhindern. Umgekehrt besteht bei zu starker Gewichtung motivationaler Aspekte die Gefahr, dass die Lehrpersonen zu wenig Fachkompetenz erlangen und sich gleichzeitig selbst überschätzen. Positiv formuliert dürften aber motivierte Lehrpersonen eher bereit sein, sich auch zukünftig mit dem Thema auseinanderzusetzen und sich weiterzubilden.

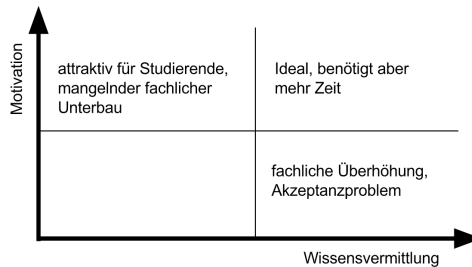


Abb. 4: Das Dilemma von Motivation vs. Wissensvermittlung bei begrenzten Zeitressourcen

5 Fazit und Empfehlungen

Informatik als Thema wird in wenigen Jahren in den Lehrplänen der meisten Deutschschweizer Kantone angekommen sein. Es stellt sich nicht mehr die Frage, »ob« sondern »wie« Informatik in der Volksschule (oder gar im Kindergarten) unterrichtet werden soll. Die Aus- und Weiterbildungen an den Pädagogischen Hochschulen stehen nun vor der Herausforderung, die angehenden und bestehenden Lehrpersonen zu befähigen, diese Inhalte auch stufenadäquat unterrichten zu können. Angesichts der in Abschnitt 1.3 beschriebenen Herausforderungen und den ersten Erfahrungen an der PHSZ empfehlen wir, in der Aus- und Weiterbildung insbesondere motivationale Aspekte gegenüber der Vermittlung umfangreichen Fachwissens zu betonen. Aufgrund der fehlenden eigenen Erfahrungen braucht es konkrete Beispiele, um sowohl eine Vorstellung zu entwickeln wie Informatik in der Volksschule aussehen kann als auch das Selbstvertrauen der Lehrpersonen zu stärken, diese Inhalte selbst umsetzen zu können. Es gilt, das eher negative Bild der Informatik mit den vorherrschenden Fehlvorstellungen zu korrigieren. Bei der Veranstaltungsplanung muss das Vorwissen der Studierenden bzw. Lehrpersonen berücksichtigt werden. Eine inhaltliche Überfrachtung und damit verbundene Überforderung könnte langfristig der flächendeckenden Einführung von Informatik in der Schule schaden.

Die Informatikdidaktik muss somit sowohl für sich selbst als auch nach außen die Erwartungen an rasche Kompetenzzuwächse im Bereich der Informatik an Schulen dämpfen. Nach “don’t miss the boat” [Ga13] ist es nun wichtig, dieses Boot nicht gleich zu überladen.

Literaturverzeichnis

- [BM05] T. Beaubouef, J. Mason: Why the High Attrition Rate for Computer Science Students: Some Thoughts and Observations, ACM SIGCSE Bull. June 2005 Vol. 37, Num. 2, S. 103-106 <https://doebe.li/t19532>
- [Bu13] Burri, A.: Jeder Schüler ein kleiner Programmierer. Tages Anzeiger vom 29.06.2013 <https://doebe.li/t15492>
- [Ga13] Gander, W. et.al.: Informatics education: Europe cannot afford to miss the boat. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education. 2013, <https://doebe.li/t14733>
- [Br16] Brinda, T. et al: Bildung in der digitalen vernetzten Welt – Dagstuhl-Erklärung. <http://tinyurl.com/dagstuhl3eck>, <https://doebe.li/t19500>
- [DEDK10] Deutschschweizer Erziehungsdirektorenkonferenz (D-EDK): Grundlagen für den Lehrplan 21, verabschiedet von der Plenarversammlung der deutschsprachigen EDK-Regionen, 2010: http://www.lehrplan.ch/sites/default/files/Grundlagenbericht_0.pdf <https://doebe.li/t11540>
- [DEDK13] Deutschschweizer Erziehungsdirektorenkonferenz (D-EDK): ICT und Medien. In: D-EDK: Lehrplan 21, Entwurfsfassung, 2013: http://konsultation.lehrplan.ch/downloads/container/30_10_0_0_1_1.pdf <https://doebe.li/t17000>
- [DEDK15] Deutschschweizer Erziehungsdirektorenkonferenz (D-EDK): Medien und Informatik. In: D-EDK: Lehrplan 21. Bereinigte Fassung, 2015. http://projekt.lehrplan.ch/lehrplan/V5/ablage/FS1E_Modul_MI.pdf <https://doebe.li/t17600>
- [Dö15] Döbeli Honegger, B.: Digitale Kompetenzen von Lehrpersonen für den Lehrplan 21, 2015. <https://doebe.li/t17550>
- [Dö16] Döbeli Honegger, B.: Mehr als 0 und 1. bern: hep Verlag, 2016. <https://doebe.li/b6000>
- [ICTS13] ICTSwitzerland: Digitale Kompetenzen benötigen Verbindlichkeit im Lehrplan 21. Positionspapier der Kommission Bildung von ICTSwitzerland, 2013. <http://tinyurl.com/verbindlichkeit>, <https://doebe.li/t17550>
- [Ec16] Eckert, N.: Eingangsbefragung 2016 - Studiengang Primarstufe der Pädagogischen Hochschule Schwyz (unveröffentlicht), 2016.
- [HSW06]: U. v. Holdt, H. Schneider, B. Wagner: Analyse von Studienverläufen und Studienabbrüchen in den Bachelorstudiengängen Informatik an der Leibniz Universität Hannover. HDI 2006 - Hochschuldidaktik der Informatik. Organisation, Curricula, Erfahrungen. 2. GI- Fachtagung 7.-8. 12. 2006 in München. <https://doebe.li/t12101>
- [KB14] Kommer, S. & Biermann, R.: Der mediale Habitus von (angehenden) LehrerInnen. In: Jahrbuch Medienpädagogik 9. Springer VS., 2014 <https://doebe.li/t14704>
- [PM14] Peyton Jones, S., Muuß-Merholz, J.: Schulfach "Computing" ab Klasse 1. In: c't 14/2014, <https://doebe.li/t16648>

Ein Modell zur Analyse von Vorstellungen über Roboter und ihrer Funktionsweise

Kathrin Müller, Carsten Schulte¹

Abstract: Kinder wachsen heute in einer Welt auf, in der sie immer früher und häufiger digitalen Artefakten begegnen. Daher wird auch die Forderung nach informatischer Bildung für Kinder in der Primarstufe immer größer. Informatische Bildung sollte Kindern ermöglichen, Informatiksysteme, deren Funktionsweise sowie unterschiedliche Rollen im Alltag wahrnehmen zu können. Als exemplarisches Informatiksystem werden hier Roboter verwendet. Doch mit welchen eigenen Vorstellungen begegnen die Kinder diesen Robotern? Bisherige Studien konzentrieren sich besonders auf Fragen wie: Sind Roboter lebendig? Sind Roboter intelligent? In diesem Artikel stellen wir ein Modell vor, welches auf Basis der Ergebnisse vorangegangener Studien und dem Ansatz der Dualitätsrekonstruktion entwickelt wurde. Mit diesem Modell ist es möglich, Vorstellungen zur Funktionsweise und Rollen von Robotern zu analysieren und klassifizieren. Es ermöglicht, einen Überblick über unterschiedliche Vorstellungen der Kinder zu bekommen und somit eine darauf abgestimmte informatische Bildung konzipieren zu können, welche die vorhandenen Vorstellungen zielgerichtet erweitern oder ggf. auch verändern kann.

Keywords: Informatik in der Primarstufe, Roboter, Schülervorstellungen

1 Einleitung

Durch die Tatsache, dass Kinder immer früher und häufiger mit digitalen Artefakten in Berührung kommen, wird auch die Forderung nach einer entsprechenden informatischen Bildung immer größer. Diese sollte es Kindern ermöglichen, Informatiksysteme, deren Funktionsweise sowie unterschiedliche Rollen im Alltag wahrnehmen zu können. Für einen anschaulichen Unterricht und einen Lebensweltbezug, wurden und werden zahlreiche Informatiksysteme für den Unterricht entwickelt. Besonders hervorzuheben sind hier Roboter. Sie existieren in unterschiedlichen Formen in der Lebenswelt der Kinder (Spielzeug, Filme, Haushalt) und bereits für die frühe informatische Bildung in KiTa und Grundschule. Doch mit welchen eigenen Vorstellungen begegnen die Kinder digitalen Artefakten und besonders diesen Robotern? Zahlreiche vorangegangene Studien (siehe Abschnitt 2) konzentrieren sich besonders auf Fragen wie: Sind Roboter lebendig? Sind Roboter intelligent, etc. Hinsichtlich eines Systemverständnisses in Verbindung mit der Programmierung des Roboters ist uns kaum Forschung bekannt. Diese Forschungslücke sollte unserer Meinung

¹ Universität Paderborn, Didaktik der Informatik, Fürstenallee 11, 33102 Paderborn, [vorname].[nachname}@uni-paderborn.de

nach im Hinblick auf die informatische Bildung geschlossen werden. Roboter stehen hier nur exemplarisch für eine wichtige Klasse von digitalen Artefakten. Andere digitale Artefakte funktionieren nach denselben oder zumindest ähnlichen Prinzipien.

Robotern wird noch eher als anderen digitalen Artefakten entsprechendes autonomes und intentionales Verhalten zugetraut (per Definition der Technologiekategorie), da sie bewusst für eine solche Wahrnehmung und solchen Einsatz (vgl. die Forschungsrichtung *social robotics*) entwickelt und entworfen werden. Arthur C. Clarke [Wi16] geht sogar soweit, dass jede hinreichend fortgeschrittene Technologie nicht von Magie zu unterscheiden sei. Wir wissen aus vielen Praxisbegegnungen, -berichten und Forschungsergebnissen, dass Computertechnologie an sich vermenschlicht wird (Elektronengehirn, ...). Es ist daher eine Aufgabe der informatischen Bildung, diese vorunterrichtlichen Alltagsvorstellungen aufzugreifen und ihnen angemessene Vorstellungen über die tatsächlichen Wirkprinzipien gegenüberzustellen. Daher bietet sich die Nutzung dieser Technologie besonders an, da ihre Faszination zu Neugier und Hinterfragen anregt, und andererseits hier die Neigung bzw. Zuschreibung von menschlichen Eigenschaften vermutlich noch größer ist. An diesem Gegenstand lässt sich also besonders gut erforschen, welche Vorstellungen Kinder haben, und mit welchen didaktischen Konzepten diese verändert bzw. angereichert werden können.

In diesem Papier stellen wir ein Modell vor, welches Ergebnisse vorangegangener Studien zu Robotervorstellungen und den Ansatz der Dualitätsrekonstruktion [Sc09] zusammenbringt. Das Modell klassifiziert und beschreibt Schülervorstellungen (deskriptiver Aspekt) und ermöglicht es, diese Vorstellungen einzuordnen und zu bewerten (normativer Aspekt). Darüber hinaus dient es als Werkzeug, um vorhandene Vorstellungen zu erweitern oder zu verändern (konstruktiver Aspekt) (vgl. die Einteilung bei [Ra11]).

2 Vorstellungen über Roboter

Im Folgenden möchten wir einen Überblick über uns bekannte Forschungen zu Vorstellungen von Robotern geben. Bei der Recherche fiel auf, dass viele dieser Studien bereits älteren Datums sind. Die Technik hat sich seitdem stark weiterentwickelt, dennoch gehen wir davon aus, dass viele dieser Vorstellungen auch heute noch von Relevanz sind. Einen guten ersten Überblick über Vorstellungen zu Computern allgemein gibt [RP16]. Auf Basis einer Literaturrecherche werden hier die Vorstellungen in *Computer sind intelligent*, *Computer sind allwissende Datenbanken*, *Computer sind mechanisch*, *Computer sind Kabelnetzwerke* und *Computer sind programmierbar* gruppiert. Besonders im Bereich *Computer sind intelligent* weist der Artikel auch auf Studien im Bereich Vorstellungen über Roboter hin. Die Ergebnisse dieser und weiterer Studien stellen wir nun im Folgenden gegliedert vor.

Psychologische und technologische Perspektiven

Nach Turkle [Tu05] und Ackermann [Ac91] gibt es psychologische und physische bzw. technologische Perspektiven. Von der psychologischen ist die Rede, wenn der Gegenstand mit lebendigen Ansichten, Emotionen, Persönlichkeit oder eigenem Willen versehen wird. Im

Gegensatz dazu sprechen sie bei technologischen Perspektiven von leblosen Materialien, wie Motoren, Sensoren und Kontrollprogrammen.

Turkle [Tu05] hat Kinder in einem natürlichen/ gewohnten Umfeld mit intelligenten Spielsachen konfrontiert und die Gespräche der Kinder aufgezeichnet. Sie fand in ihrer Studie sowohl eine eher psychologische Sichtweise - wie z.B. ein sprechendes Spielzeug ist lebendig, da sprechende Wesen automatisch lebendig sind - aber auch eine gegenteilige Sichtweise auf intelligente Spielsachen heraus. Die gegenteilige Sichtweise behauptet, lebendige Wesen müssen mit Intentionen handeln, daher könne ein sprechendes Spielzeug nicht lebendig sein.

Ackermann [Ac91] geht von zwei ähnlichen Perspektiven aus: „psychologisch“ und „engineering“. Beim psychologischen Bild werden intelligente Artefakte als lebendig angesehen, wenn sie Intentionen, Bewusstsein, Persönlichkeit oder einen eigenen Willen zu haben scheinen. Die Perspektive „engineering“ wird typischerweise benutzt, wenn das Bild eines gebauten oder programmierten Systems vorherrscht, dass das System gebaut oder programmiert wird und keine Intentionen aufweist, sondern das Verhalten lediglich durch die Interaktion der einzelnen Komponenten zustande kommt. Laut Ackermann ist eine vollständige Beschreibung jedoch nur durch beide Perspektiven möglich: Die Fähigkeit, Objekte zu beleben bzw. Objekten Leben zu geben, sei ein entscheidender Schritt in Richtung des Aufbaus kybernetischer Theorien und kein Zeichen von kognitiver Unreife. Es wird die Fähigkeit erlangt, Perspektiven gezielt zu wechseln und für den richtigen Zweck einzusetzen. Daher sollte es ein Ziel informatischer Bildung sein, diese Fähigkeit zu fördern. Levly und Mioduser [LM08] bestätigen diese Perspektiven in einer Studie mit fünf- und sechsjährigen Kindern in gewisser Weise. Die Studie untersucht die Erklärungsperspektiven hinsichtlich eines mobilen Roboters, während die Kinder sein Verhalten anhand von Regeln programmieren. Als Ergebnis lassen sich zwei Typen von Schülerperspektiven finden: Die Perspektive „bridging“ stellt dabei eine Mischperspektive aus „engineering“ und der psychologischen Perspektive dar. Außerdem wird im Gegensatz zu anderen Studien, die die Kinder bzw. Schülerinnen und Schüler ausschließlich die Roboter beobachten lassen, deutlich, dass die Kinder durch die Steuerung von Robotern eine Vorstellung davon entwickelt haben, wie Roboter funktionieren und arbeiten.

Resnick und Martin [RM91] unterteilen die wechselnden Perspektiven in der Grundschule beim Umgang mit Lego Robotern in psychologische (Tier/ lebendig), mechanische (Maschinen) und informatische (Informationsfluss von einem Teil zum anderen) Denkmuster.

Eigenschaften von Robotern

In den im vorherigen Abschnitt vorgestellten Studien lassen sich bereits erste Eigenschaften erkennen, die Kinder Robotern zuweisen.

Bernstein und Crowley [BC08] beschäftigten sich mit der Frage, inwieweit Erfahrungen mit Robotern und intelligenten Systemen im Kindesalter die Konzepte von „lebendig sein“ und „Intelligenz“ beeinflussen. Um dies zu untersuchen, wurden 60 Kinder zwischen 4 und 7 Jahren über intellektuelle, psychologische und biologische Eigenschaften von acht Dingen befragt. Zusätzlich wurden die Eltern der Kinder befragt. Themen waren hier: Zugänglichkeit von Robotern und digitalem Spielzeug und/oder Lernmaterial, Roboter im

Haushalt und Interesse und Vorwissen über Roboter und Computer. Daraus ergab sich ein quantitatives Maß der Vorerfahrungen der Kinder.

In dieser Studie wurde deutlich, dass einem Roboter im Bereich der Intelligenz mehr Eigenschaften zugeordnet wurden als einem Computer. Auf der Ebene der Intelligenz sind die Werte vergleichbar mit einer Katze. Die Werte für biologische Eigenschaften lagen im mittleren Segment, vergleichbar mit einer Pflanze. In der durchschnittlichen Bewertung der psychologischen Eigenschaften wurde deutlich, dass gut die Hälfte der Kinder Robotern einen freien Willen und/oder eigene Emotionen zuschreiben (Werte lagen im Mittelfeld). In [Be11] werden weitere kognitive Eigenschaften wie z.B. Erinnerungsvermögen genannt. Darüber hinaus werden hier emotionale und verhaltensorientierte Eigenschaften genannt.

Roboter und Animismus

In den vorherigen Abschnitten wurde bereits deutlich, dass es Kinder gibt, die Robotern menschliche Charakteristika wie beispielsweise eine Persönlichkeit oder Bewusstsein oder gar Lebendigkeit zuordnen. Diese Tendenz, leblosen Objekten menschliche Charakteristika zuzuordnen wurde ursprünglich von Piaget entdeckt und wird als Animismus bezeichnet (vgl. [Be11]). Es gibt zahlreiche Studien, in denen der mehrdeutige Status von technischen Objekten untersucht wird. Einige wie z.B. [DS96], [Be11] oder [BC08] zeigen, dass der beschriebene Animismus häufig auch in Bezug auf Roboter vorliegt.

Kategorien von Robotern

Im Rahmen der Studie von Bernstein und Crowley [BC08] wurden aus den bereits zuvor vorgestellten Eigenschaften von Robotern drei Kategorien zur Klassifizierung von Robotern herausgearbeitet. Dabei handelt es sich um die Kategorien Roboter als Tier, Roboter als Maschine und Roboter als intelligente Technik (smart technology). Roboter als Tiere weisen die Eigenschaft Lebendigkeit sowie viele psychologische und kognitive Eigenschaften auf. Roboter als Maschine weisen keine Eigenschaft im Bezug auf Lebendigkeit auf. Psychologische und kognitive Eigenschaften sind kaum bis gar nicht vorhanden. Im Zuge der Kategorie Roboter als intelligente Technik machen die Autoren keine Aussage zur Eigenschaft Lebendigkeit. Die Kategorie weist aber viele kognitive Eigenschaften auf und Kinder, deren Vorstellungen in diese Kategorie eingeordnet werden, haben bereits mehr Erfahrungen oder Begegnungen mit Robotern gehabt.

Aus der Literaturanalyse wird deutlich, dass der Roboter häufig als technologische Blackbox betrachtet wird. Der Fokus liegt auf den biologischen, psychologischen und intellektuellen Eigenschaften sowie den Funktionen des Roboters. Lediglich die technologische Perspektive betrachtet Motoren, Sensoren etc. und bietet somit Hinweise auf ein technisches Verständnis der Roboter. Um ein vollständigeres Bild über Roboter im Kontext der informatischen Bildung zu erhalten, führen wir im folgenden Abschnitt eine Dualitätsrekonstruktion dieses digitalen Artefakts durch.

Dualitätsrekonstruktion und Roboter

Die Dualitätsrekonstruktion [Sc09] ist ein Verfahren zur didaktischen Analyse von digitalen Artefakten bzw. Informatiksystemen für die informatische Bildung. Es verschränkt die

Beschreibung von Präkonzepten und angestrebten Kompetenz- bzw. Lernzielen mit der exemplarischen Analyse der wesentlichen informatischen Aspekte des Unterrichtsgegenstands. Diese sind generell die Funktionsweise (*Struktur*) sowie die mit dem System verknüpften Absichten, Einsatzziele, -zwecke und Nebenwirkungen (*Funktion*). Um diese zu fassen, wird jeweils auch auf die *technikgeschichtliche Entwicklung* der betrachteten Klasse von Informatiksystemen eingegangen.

In den 1940er prägte Isaac Asimov in seinen Science-Fiction-Geschichten die berühmten drei (später auf vier erweiterten) Robotergesetze, denen zufolge ein Roboter keinen Menschen (bzw. später die Menschheit) schädigen darf, den Menschen gehorchen und sich selbst schützen muss - jeweils soweit nicht ein übergeordnetes Gesetz dagegen spricht. Tatsächlich gebaut wurden die ersten Roboter in den 1960er und 1970er Jahren, zunächst als Master-Slave-Arm zum Umgang mit atomaren Stoffen, später dann als sogenannte Industrie-Roboter in der industriellen Produktion. In den 1980er Jahren ging es um die intelligente Verknüpfung von Wahrnehmung und Operation: Durch ausgeklügelte programmierte Steuerung sollten Roboter anhand ihrer Sensoren die Umgebung erfassen und dementsprechend die Umgebung manipulieren oder sich selbst darin bewegen (mittels Aktoren). Mit dem Fortschreiten der Technologie erweiterte sich der Einsatzbereich von stark begrenzten und kontrollierbaren Umgebungen (Fertigungsstraße, Operationsaal, etc.) zu alltäglichen Umgebungen (Haushalt,...), und damit auch die Interaktion zwischen Mensch und Roboter von der engen Kontrolle zu stärkerer Autonomie und von der strikten Trennung der Einsatzbereiche bzw. -orte zu kooperierenden Formen, sodass heute auch von social robotics und co-bots anstelle von ro-bots gesprochen wird.

Die *Funktion* von Robotern wird daher auch mit Automatisierung (im Sinne Objektivierung: Übertragen von Handlungen an eine Maschine, die diese dann ausführt) nicht vollständig erfasst. Roboter werden eher menschenähnlich als Teil der Handlungen (z.B. Arbeitsprozesse) bzw. als autonom agierende Wesen betrachtet, die zur sozialen Umgebung dazugehören (z.B. Roboter in der Pflege). Anders formuliert: Robotern wird noch stärker als anderen digitalen Artefakten Autonomie und Interaktion als eigenständige Handlungsfähigkeit in einer sozialen Umgebung zugesprochen. Damit wird das Verhalten einer Maschine tendenziell als menschenähnlich, intentional aufgefasst.

Auf der Betrachtungsebene der *Struktur* wird deutlich, dass dieses (emergente) Verhalten auf einer Kombination von Sensoren, Programmsteuerung und Aktoren beruht. Sensoren umfassen typischerweise optische Sensoren, aber auch akustische und Berührungssensoren sowie weitere „Sinne“, die Menschen so nicht haben (Radar, Ultraschall, u.a). Aktoren dienen der Bewegung im Raum (Räder, u.a.) und der Manipulation von Gegenständen bzw. der Umgebung (Greifarm, u.a.).

3 Modell zur Analyse von Vorstellungen über Roboter und ihre Funktionsweise

Im Folgenden stellen wir nun unser theoretisches Modell zur Analyse und Einordnung von Vorstellungen über Roboter und ihrer Funktionsweise vor. Um ein möglichst vollständiges Bild von Robotern in unserem Modell abzubilden, haben wir die Ergebnisse der uns bekannten Studien um weitere Charakteristika und um die zusätzlichen Perspektiven Struktur und Funktion aus der Dualitätsrekonstruktion ergänzt. Abbildung 1 zeigt unser

Perspektive Merkmale	technologisch	psychologisch	Struktur	Funktion
biologisch				
mechanisch/technisch				
emotional				
kognitiv/intellektuell				
intentionales Verhalten				
steuerbar (manuell, automatisch)				

Klassifizierung
Roboter als Lebewesen (soziales Wesen)
Roboter als Maschine
Roboter als intelligente Technik (smart technology)
Roboter als soziale Technik

Abb. 1: Modell zur Analyse und Einordnung von Vorstellungen über Roboter und ihre Funktionsweise
vollständiges Modell bestehend aus Perspektiven und Merkmalen von Robotern. Ergänzt wird das Modell um eine Klassifizierung von Robotern. In den folgenden Abschnitten beschreiben wir die einzelnen Aspekte des Modells und die vorgenommenen Ergänzungen genauer.

Perspektiven auf Roboter

In der Literatur wurde deutlich, dass Kinder Roboter aus den Perspektiven technologisch oder psychologisch, bzw. einer Mischung der beiden Perspektiven beschreiben (vgl. z.B. [LM08]). Wie bereits erwähnt, wurde in den vergangenen Studien wenig Fokus auf die technologische Funktion und Struktur von Robotern gelegt. Würde man nur - beispielsweise - die psychologische Perspektive mit der technologischen Perspektive ergänzen oder ersetzen, fehlt ein wichtiger Aspekt informatischer Bildung, der leider oft übersehen wird. Verstehen einer Technologie ist nicht reduzierbar auf das Verstehen eines Artefakts oder eines Phänomens an sich. Es lässt sich auch nicht auf die Analyse und Beschreibung reduzieren. Denn dann würde man die Technologie quasi naturwissenschaftlich als gegebenen Untersuchungsgegenstand - als naturgegeben - hinnehmen. Dabei ist der Unterschied zwischen Technologie und Natur, dass die Technologie und damit technische Artefakte von Menschen mit bestimmten Absichten künstlich erschaffen worden sind und weiterhin neue erfunden und erschaffen werden. Diese beiden Perspektiven werden daher als Struktur (Aufbau eines technischen Systems) und Funktion (Interpretation eines technischen Systems) ergänzt. Zusammenfassend umfasst das Modell die folgenden Perspektiven:

technologisch: Diese Perspektive betrachtet technische Bauteile und Eigenschaften und bietet somit Hinweise auf ein technisches Verständnis der Roboter.

psychologisch: Der Fokus dieser Perspektive liegt auf den biologischen, psychologischen und intellektuellen Eigenschaften sowie den Funktionen des Roboters.

Struktur: Diese Perspektive charakterisiert sich durch beobachtbare Merkmale, die messbar und objektiv beschreibbar sind.

Funktion: Diese Perspektive charakterisiert eine interpretierende Zuschreibung: Mehr oder weniger plausible Annahmen über Effekte, Nutzen, Angemessenheit, Einsatzzweck etc..

Merkmale von Robotern

Die Merkmale von Robotern in unserem Modell setzen sich aus den in der Literatur bereits benannten Merkmalen biologisch, emotional, kognitiv sowie intentionalem Verhalten und der Ergänzung um mechanisch/technisch und steuerbar zusammen. Die Ergänzung wurde vorgenommen, um im Bereich der Merkmale eine wichtige Komponente aus den Bereichen der Funktionsweise und Struktur hinzuzufügen. Konkret sind die einzelnen Merkmale wie folgt zu betrachten:

biologisch: Hinter biologischen Merkmalen verbergen sich neben dem Animismus, also der Zuschreibung von Lebendigkeit, weitere Aspekte, die Lebewesen zugeschrieben werden können, wie z.B. Wachsen oder Atmen sowie Sinne.

mechanisch/technisch: Hinter mechanischen, bzw. technischen Merkmalen verbergen sich insbesondere mechanische, bzw. technische Komponenten wie z.B. Sensoren und Aktoren und entsprechende mechanisch/technische Eigenschaften und Abläufe.

emotional: Emotionale Merkmale sind solche, die man dem Bereich Emotionen und dem emotionalen Erleben zuordnen kann. Hierzu gehören aber auch Interessen und Einstellungen.

kognitiv/intellektuell: Kognitive bzw. intellektuelle Merkmale beinhalten alle kognitiven bzw. intellektuellen Fähigkeiten wie beispielsweise Denken, Rechnen, etwas wissen etc.

intentional: Unter intentionalem Verhalten werden alle Verhaltensweisen betrachtet, die zielgerichtet, bzw. beabsichtigt sind. Hierzu gehört auch Sprechen.

steuerbar: Merkmale im Bereich steuerbar gliedern sich auf in direkt und indirekt steuerbar. Unter direkt steuerbar wird beispielsweise die Steuerung über Hebel oder eine Fernbedienung verstanden. Indirekt steuerbar beinhaltet die Programmierung.

Klassifizierung von Robotern

Zur Klassifizierung von Robotern übernehmen wir die drei Kategorien aus [BC08] mit der Ergänzung um Roboter als soziale Technik und definieren diese wie folgt:

Roboter als Lebewesen: In diese Kategorie fallen Vorstellungen über Roboter, die diese als lebendig ansehen und darüber hinaus neben eventuellen weiteren biologischen Merkmalen, Merkmale aus der Struktur-Perspektive aufweisen. Aus psychologischer Perspektive gibt es Merkmale im Bereich der Emotionalität, der Kognition sowie dem intentionalen Verhalten. Diese Merkmale können zusätzlich auch aus der Funktions-Perspektive auftauchen.

Roboter als Maschine: In diese Kategorie fallen Vorstellungen über Roboter, die diese

nicht als lebendig ansehen, also keine biologischen Merkmale aus der Struktur-Perspektive aufweisen. Aus Struktur- sowie technologischer Perspektive weisen Vorstellungen dieser Kategorie mechanische bzw. technische Merkmale auf. Diese sind z.B. aus Struktur-Perspektive technische Bauteile. Darüber hinaus weisen Vorstellungen dieser Kategorie aus Funktions-Perspektive Merkmale im Bereich steuerbar auf.

Roboter als intelligente Technik: Vorstellungen über Roboter als intelligente Technik weisen eine Abwesenheit von biologischen und emotionalen Merkmalen in allen Perspektiven auf. Im Prinzip erweitern Vorstellungen dieser Kategorie die Kategorie Roboter als Maschine um kognitive Merkmale aus technologischer und Funktions-Perspektive.

Roboter als soziale Technik: Vorstellungen dieser Kategorie weisen im Unterschied zu Robotern als intelligente Technik Merkmale aus dem Bereich der Emotionalität aus Funktions- und technologischer Perspektive auf. Kognitive Merkmale sind ggf. gar nicht oder geringer vorhanden.

Abbildung 2 zeigt die Einordnung der einzelnen Kategorien in das Modell.

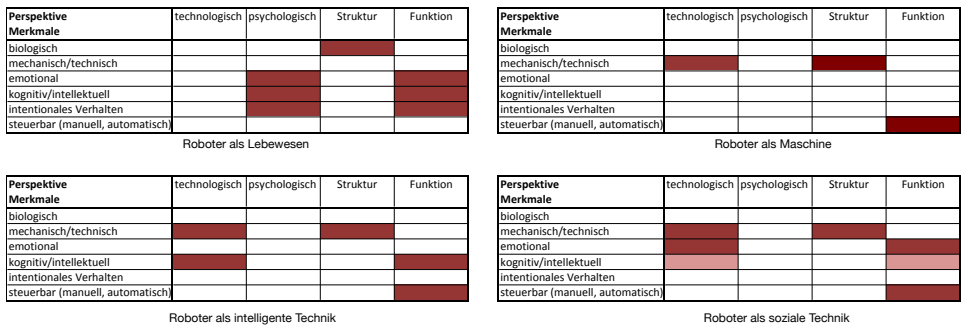


Abb. 2: Klassifizierung von Robotervorstellungen anhand des Modells

3.1 Anwendungsfelder

Die Anwendungsfelder für das hier vorgestellte Modell lassen sich in normative, deskriptive und konstruktive Aspekte unterteilen. Der deskriptive Aspekt umfasst dabei die Kategorisierung der Vorstellungen und die Einordnung der einzelnen Vorstellungsbestandteile in das Modell. Dabei ist die Form der Vorstellungserhebung z.B. Interview, Text, Bild etc. nicht von Relevanz. Die erhobenen Vorstellungen können zunächst z.B. mit einer qualitativen Inhaltsanalyse ausgewertet werden. Als Kategorien dienen hierbei die einzelnen Perspektiven und Merkmale des Modells (vgl. Abb. 1, Tabelle links). Die Ergebnisse können dann anhand der vorgegebenen Klassifizierung eingeordnet werden (vgl. Abb. 1, Auflistung rechts). Auf diese Art und Weise können sowohl Vorstellungen von einzelnen Individuen als auch von Gruppen analysiert und beurteilt werden. Die so gesammelten Vorstellungen werden sichtbar und können damit in Lernprozessen berücksichtigt werden.

In einem weiteren Anwendungsfeld (normativer Aspekt) kann untersucht werden, inwieweit die vorhandenen Vorstellungen ausreichend bzw. hilfreich sind. Hierzu wird zunächst festgelegt, wie Roboter zu verstehen sein sollten. Angemessene Vorstellungen sind unserer Meinung nach eine Verknüpfung der Kategorien *Roboter als intelligente Technik* und *Roboter als soziale Technik* (vgl. Abb. 2). Offen ist, inwiefern die psychologische Perspektive dabei weiterhin berücksichtigt werden soll und ob die technologische Perspektive feiner unterteilt werden sollte [Ac91; RM91].

Im dritten Anwendungsfeld, dem konstruktiven Aspekt, können basierend auf bekannten Schülervorstellungen und normativen Modellen Methoden entwickelt und evaluiert werden, die zur Erweiterung und ggf. Veränderung der vorhandenen Vorstellungen beitragen. Wir können mit unserem Modell die vorunterrichtlichen Vorstellungen wie folgt beschreiben: Sie sind vermutlich im Bereich *Roboter als Lebewesen*, bzw. *Roboter als Maschine* angesiedelt (Abb. 2 oben). Aus normativer Perspektive angemessene Vorstellungen wären aus den Bereichen *Roboter als intelligente*, bzw. *soziale Technik* (Abb. 2 unten). An der Gegenüberstellung mit dem Modell kann nun direkt abgelesen werden, auf welchen Vorstellungen im Unterricht aufgebaut werden kann. So können Konzepte und Werkzeuge entstehen, die es ermöglichen Vorstellungen von Lernenden zu diagnostizieren und sinnvoll zu verändern, bzw. anzureichern. Für die Erforschung von didaktischen Konzepten und Lernprozessen kann das Modell an unterschiedlichen Stellen als Werkzeug zur Einordnung der Vorstellungen eingesetzt werden.

4 Zusammenfassung und Ausblick

Dieser Artikel stellt neben einem Überblick zum Stand der Forschung zu Robotervorstellungen ein Modell vor, welches ein Werkzeug zur Analyse und Einordnung von Vorstellungen über Roboter und ihrer Funktionsweise darstellt. Dieses theoretisch erstellte Werkzeug wird in einem nächsten Schritt in der Praxis eingesetzt, um Vorstellungen von Grundschulkindern zu Robotern zu analysieren und einzuordnen. Dabei wird es empirisch evaluiert und ggf. um fehlende Komponenten ergänzt. Eine interessante Fragestellung an dieser Stelle ist, ob die Perspektive *technologisch* im Anwendungskontext ausreichend ist, oder, ob sie wie bei [RM91] in mechanische und informatische Perspektive unterteilt werden sollte.

Durch die Einordnung von Vorstellungen in das Modell ist es möglich, Hypothesen zur Beeinflussung der Vorstellungen zu generieren. Hier sind unterschiedliche Wege vorstellbar, wie z.B. mit Konzepterweiterung und Konzeptwechsel. Darüber hinaus ist es interessant zu erforschen in wie weit in der Kategorie *Roboter als intelligente Technik* und *Roboter als soziale Technik* psychologische Vorstellungen vorhanden sein sollten, um ein umfassendes Verständnis zu Robotern und ihrer Funktionalität zu besitzen. Dies ist gerade dann relevant, wenn es darum geht, Roboter in unterschiedlichen Rollen wahrnehmen zu können und ihnen auch unterschiedliche Rollen zuweisen zu können. Ob dies ohne eine psychologische Perspektive überhaupt möglich, oder wir unsere Kategorien *Roboter als intelligente Technik* und *soziale Technik* um Merkmale aus psychologischer Perspektive ergänzen müssen, ist hier eine spannende Frage.

Wie bereits in der Einleitung erwähnt, dienen Roboter hier als exemplarisches Beispiel für digitale Artefakte. Es ist daher interessant zu untersuchen, welche Veränderungen am Modell für andere digitale Artefakte notwendig sind.

Literatur

- [Ac91] Ackermann, E.: The agency model of transactions: Toward an understanding of children's theory of control. *Psychologie genetique et sciences cognitives*. Geneve: Fondation Archives Jean Piaget/, 1991.
- [BC08] Bernstein, D.; Crowley, K.: Searching for Signs of Intelligent Life: An Investigation of Young Children's Beliefs About Robot Intelligence. *Journal of the Learning Sciences* 17/2, S. 225–247, 2008.
- [Be11] Beran, T. N.; Ramirez-Serrano, A.; Kuzyk, R.; Fior, M.; Nugent, S.: Understanding how children understand robots: Perceived animism in child–robot interaction. *International Journal of Human-Computer Studies* 69/7–8, S. 539–550, 2011.
- [DS96] van Duuren, M.; Scaife, M.: “Because a robot's brain hasn't got a brain, it just controls itself” — Children's attributions of brain related behaviour to intelligent artefacts. *European Journal of Psychology of Education* 11/4, S. 365, 1996.
- [LM08] Levy, S. T.; Mioduser, D.: Does it “want” or “was it programmed to...”? Kindergarten children's explanations of an autonomous robot's adaptive functioning. *International Journal of Technology and Design Education* 18/4, S. 337–359, 2008.
- [Ra11] Rabel, M.: Grundvorstellungen in der Informatik. In (Weigend, M.; Thomas, M.; Otte, F., Hrsg.): *Informatik mit Kopf, Herz und Hand. Praxisbeiträge zur INFOS 2011*. ZfL-Verlag, Münster, S. 61–70, Sep. 2011, ISBN: 978-3-86877-009-4.
- [RM91] Resnick, M.; Martin, F.: Children and artificial life. In (Hartel, I.; Papert, S., Hrsg.): *Constructionism*. Ablex, S. 41–71, 1991.
- [RP16] Rücker, M. T.; Pinkwart, N.: Review and Discussion of Children's Conceptions of Computers. *Journal of Science Education and Technology* 25/2, S. 274–283, 2016.
- [Sc09] Schulte, C.: Dualitätsrekonstruktion Als Hilfsmittel Zur Entwicklung Und Planung von Informatikunterricht. In (Koerber, B., Hrsg.): *Zukunft Braucht Herkunft. INFOS'09*. S. 355–366, 2009, ISBN: 978-3-88579-250-5.
- [Tu05] Turkle, S.: *The Second Self: Computers and the Human Spirit* (MIT Press). The MIT Press, 2005.
- [Wi16] Wikipedia: Clarkesche Gesetze. In: Wikipedia. Page Version ID: 158717013, 13. Okt. 2016, URL: https://de.wikipedia.org/w/index.php?title=Clarkesche_Gesetze&oldid=158717013, Stand: 19. 01. 2017.

Schülervorstellungen im Zusammenhang mit Smartphones

Torsten Brinda¹ und Friederike Braun²

Abstract: Smartphones haben einen rasanten Aufstieg hinter sich. Von den ersten erschwinglichen Mobiltelefonen bis zu modernen High-Tech-Geräten sind sie nicht nur komplexer, sondern auch immer beliebter geworden. Im Jahr 2016 besaßen fast 100% aller Jugendlichen in Deutschland ein eigenes Smartphone und benutzten es ständig. Smartphones weisen für Lernende in Schulen einen hohen Lebensweltbezug auf und eignen sich daher für eine unterrichtliche Betrachtung als Informatiksystem auch im Hinblick auf dessen Wirkung auf Individuum und Gesellschaft, weiterhin zur Anzeige schulbezogener Medieninhalte. Für eine didaktische Aufbereitung des Gegenstandes „Smartphone“ für den Unterricht mangelt es jedoch unter anderem noch an wissenschaftlichen Arbeiten zu diesbezüglichen Schülervorstellungen. Deshalb wurde eine explorative Studie zur Erfassung von Schülervorstellungen im Zusammenhang mit Smartphones durchgeführt. Aus acht halbstrukturierten Interviews wurde ein erster Überblick über bestehende Vorstellungen zu Teilaspekten von Smartphones gewonnen.

Keywords: Schülervorstellungen, Smartphone, halbstrukturierte Interviews, explorative Studie.

1 Einleitung

In unserer modernen Welt sind wir in beständig zunehmenden Umfang von verschiedenen Informatiksystemen umgeben. Informatische Bildung in Schulen soll sich mit den Prinzipien und Gestaltungsgrundlagen solcher Systeme auseinandersetzen und die komplexen Wirkprozesse auf Individuum und Gesellschaft in den Blick nehmen – auch unter Einbeziehung von Informatiksystemen aus der Lebenswelt der Schülerinnen und Schüler. Schon bei einer oberflächlichen Suche sticht dabei ein System aus der Menge heraus: das *Smartphone*. Laut der JIM-Studie 2016 (vgl. mpfs.de) besitzen 95% der Jugendlichen im Alter zwischen 12 und 19 Jahren im Jahr 2016 ein Smartphone mit Touchscreen und Internetzugang. Die überwiegende Mehrheit benutzt das Gerät täglich, sei es zum Kommunizieren, zum Recherchieren im Internet, zum Hören von Musik oder zum Schauen von Videos. Für eine unterrichtliche Aufbereitung nach dem Prinzip der didaktischen Rekonstruktion [Di11] mangelt es jedoch gegenwärtig noch an empirisch gesicherten Erkenntnissen zu Alltagsvorstellungen, die Schülerinnen und Schüler durch ihren Umgang mit dem Gerät entwickeln und in den Unterricht mitbringen. Die empirische Exploration solcher Vorstellungen war daher Gegenstand einer Interviewstudie [Br16], von der in dieser Arbeit berichtet wird.

¹ Universität Duisburg-Essen, Didaktik der Informatik, Schützenbahn 70, 45127 Essen, torsten.brinda@uni-due.de

² Universität Duisburg-Essen, Institut für Informatik und Wirtschaftsinformatik, Schützenbahn 70, 45127 Essen, friederike.braun@stud.uni-due.de

2 Verwandte Arbeiten

Es wird bereits seit einiger Zeit in den Blick genommen, die Attraktivität von Smartphones unter Jugendlichen für die informatische Bildung in Schulen nutzbar zu machen. So liegen Konzepte zur Implementierung von Apps im Unterricht vor (z. B. [Br14]), weiterhin wurde damit experimentiert, Smartphones als alleinige Informatiksysteme im Informatikunterricht einzusetzen (z. B. [Mü11]). Darin zeigte sich, dass das möglich ist und dass damit einige didaktische Vorteile verbunden sein können, wie z. B. eine effektivere Nutzung der Unterrichtszeit und ein geschlechtergerechterer Zugang zum verwendeten Informatiksystem. Schülerorientierter Unterricht, der von bestehenden Schülervorstellungen ausgeht, kann bspw. nach dem Modell der didaktischen Rekonstruktion gestaltet werden, das Diethelm et al. in die Informatik übertrugen [Di11]. Ferner erweiterten sie es um gesellschaftliche Ansprüche an das zugeordnete Schulfach, die Auswahl von informatischen Phänomenen und die Einbeziehung von Lehrerperspektiven. In diesen drei Bereichen sowie den Schülervorstellungen im Informatikbereich bestehe noch Forschungsbedarf. Es existieren bisher eine Reihe von Arbeiten zu Schülervorstellungen, die einzelne Teilaspekte des vernetzten Informatiksystems Smartphone berühren, bspw. wurden Schülervorstellungen zum Aufbau und zur Funktionsweise des Internets in einer Reihe von Arbeiten erfasst (z. B. [Pa05], [DWZ12]). Informatikbezogene Arbeiten, die sich explizit auf Schülervorstellungen im Zusammenhang von Smartphones beziehen, wurden in einer ausführlichen Literaturrecherche der Autoren bislang nicht gefunden.

3 Forschungsmethodik

Zur Datenerhebung fiel die Wahl auf leitfadengestützte Interviews. Für deren Hauptphase wurden Fragen zu Kategorien entwickelt, die sich an von Desktop-Computern abgrenzenden Aspekten von Smartphones orientierten (K1 bis K4, vgl. Tab. 1).

K1: Drahtlose Verbindungen von Smartphones

K1.1: Netzwerkstruktur von Mobilfunknetzen
 K1.2: Adressierung in Mobilfunknetzen
 K1.3: Vom Smartphone genutztes Übertragungsmedium
 K1.4: Handover bei Smartphones
 K1.5: Internetprotokoll als Grundlage netzbasierter Dienste

K2: Apps auf Smartphones

K2.1: Definition einer App
 K2.2: Programmaufruf auf einem Smartphone
 K2.3: Herkunft von Apps
 K2.4: Installation von Apps

K3: Steuerung von Smartphones mit Hilfe des Touchscreens

K3.1: Bedienelemente eines Smartphone
 K3.2: Touchscreen eines Smartphones
 K3.3: Programmarchitektur eines Smartphones

K4: Kompakte Bauart von Smartphones

K4.1: In Smartphones vertretene kompakte Hardware
 K4.2: Technische Grenzen von Smartphones

Tab. 1: Deduktiv entwickeltes Kategoriensystem für Schülervorstellungen zu Smartphones

Im Herbst 2016 wurden zwölf Interviews mit Lernenden der Jgst. 9 und 11 (jeweils zur Hälfte mit und ohne Informatikunterricht) an einem Gymnasium im Ruhrgebiet in NRW durchgeführt und audioaufgezeichnet, davon acht für die Auswertung ausgewählt und mittels qualitativer Inhaltsanalyse nach Mayring ausgewertet.

4 Ergebnisse der Studie

Im Bereich **drahtloser Netzverbindungen** (K1) waren sich alle Teilnehmenden bewusst, dass ihre Smartphones an ein weitläufiges Netzwerk angeschlossen sind (K1.1). Vier Schüler stellten sich vor, dass die Verbindung über Masten erfolge, drei über Server und drei weitere über Satellit. Dass zur Adressierung von Daten eine Nummer verwendet werde, äußerten fünf von acht Teilnehmenden (K1.2). Ein Teilnehmer sagte explizit, dass die Nummer eindeutig sei, bei anderen ergab sich dies implizit aus den gewählten Beispielen (z. B. Telefonnummer), was dem Konzept der MAC-Adressen nahekommt. Eine übergreifende Vorstellung, welches Übertragungsmedium ein Smartphone nutzt, wurde nicht gefunden (K1.3). Funk wurde auf Anfrage nicht erwogen, jedoch von einem Schüler später im Kontext eines Funkloches beiläufig erwähnt. Vier von acht Schülern nannten lediglich „Signale“ als Medium. Alle Teilnehmenden waren sich bewusst, dass die Güte einer Netzwerkverbindung schwanken kann (K1.4). Die Hälfte meinte, dass unterschiedliche Dienste auch unterschiedliche Anforderungen an Netzwerkverbindungen stellen (K1.5). Drei Schüler sagten, dass die verschiedenen Dienste dasselbe Netz benutzen. Eine Schülerin schlussfolgerte, dass „alles über das Internet funktioniert“ [Br16, S. 134], weil ihre Messenger-Apps nur mit einer aktiven Internetverbindung nutzbar seien. Unter den Vorstellungen zu **Apps** (K2) sticht vor allem heraus, dass sechs der Teilnehmenden Programme auf dem Smartphone für eine bestimmte Aufgabe als Apps betrachten (K2.1). Sie definieren damit Anwendungsprogramme, auch wenn sie die Begriffe „Anwendung“ und „Application“ deutlich seltener nutzen. Fünf Schüler betrachten alles, was aus einem AppStore heruntergeladen werde, als eine App. Fünf von acht Schülern stellten sich vor, dass das Smartphone oder etwas darin beim Aufruf einer App arbeite (K2.2). Zwei präzisierten, dass Signale versendet würden. Bezüglich der in einer App enthaltenen Daten fanden sich bei zwei Schülern detailliertere Vorstellungen: Einer stellte sich vor, dass beim Aufruf einer App ein „Layout“ aufgerufen werde, das dann mit aus dem Internet geladenen Daten aufgefüllt werde. Er verglich diese „App-Basis“ mit dem in Java verwendeten Konzept des Basiskonstruktors für Klassen. Eine Schülerin gab an, dass Informationen, welche entweder vom Entwickler der App einprogrammiert oder bei vorherigen Aufrufen der App gespeichert worden seien, geladen werden. Alle Schüler hatten schon einmal eine App aus einem Store geladen (K2.3), doch nur eine Schülerin gab auch die Möglichkeit an, dass man selbst eine App entwickeln könne. Für die Hälfte der Teilnehmenden war eine App nach dem Download direkt nutzbar, weitere Installationsschritte wurden nicht erwogen (K2.4). Drei Befragte gaben an, dass Installationsschritte vor der Nutzbarkeit erforderlich seien. Hinsichtlich der **Bedienung per Touchscreen** (K3) beschränkten sich die Antworten in K3.1 fast ausschließlich darauf, welche Bedienelemente bei einem Smartphone gegeben seien. Fünf der Teilnehmenden benannten den Touchscreen als Peripheriegerät, das Berührungen wahrnehme und entsprechend Signale an eine zentrale Recheneinheit sende (K3.2). Sechs der Jugendlichen stellen sich weiterhin vor, dass das Smartphone nach vorprogrammierten Mustern auf Eingaben reagiere (K3.3). Die **kompakte Bauart** (K4) erklärte sich die Hälfte der Lernenden damit, dass die in ihnen enthaltenen Daten wenig bzw. keinen Platz wegnähmen (K4.1). Zwei konkretisierten, dass eigentlich keine Daten ge-

speichert werden müssten, da diese auf externen Speichern, wie Servern oder im Himmel fliegenden Clouds, hinterlegt würden. Zwei weitere äußerten, dass die kleinen internen Speicher in der Lage seien viele Daten aufzunehmen. Ein Schüler äußerte zudem, Daten würden keinen Platz wegnehmen, weil sie nur aus Bits und Bytes bestünden und somit nicht physisch präsent seien. Im Vergleich zu Desktop-Computern schätzten fünf Teilnehmende die Leistungsfähigkeit von Smartphones geringer ein (K4.2) mit Verweis auf den geringeren Speicherplatz, langsamere Prozessoren bzw. schlechtere Grafikkarten. Außerdem könnten weniger Peripheriegeräte angeschlossen werden.

5 Fazit und Ausblick

Im Rahmen der Studie konnten eine Reihe von Schülervorstellungen zu einzelnen Aspekten von Smartphones ermittelt werden. Dabei wurden sowohl Vorstellungen erfasst, die von einer Mehrzahl der Jugendlichen vertreten wurden, als auch solche, die nur von einzelnen Teilnehmenden genannt wurden, dafür aber deutlich detaillierter ausfielen. Nachfolgend ist es nun erforderlich, die Erfassung diesbezüglicher Vorstellungen in weiteren Arbeiten zu vertiefen und zu prüfen, inwieweit sich diese auch bei anderen Schülergruppen finden lassen. Weiterhin sollte dabei informatikspezifisches Vorwissen genauer ermittelt und kontrolliert werden, um den Einfluss von Informatikunterricht auf die Schülervorstellungen ermitteln zu können. Offen ist außerdem die Erfassung diesbezüglicher Lehrervorstellungen, wie in [Di11] angeregt. Schlussendlich sollten Empfehlungen für den Informatikunterricht abgeleitet werden.

Literaturverzeichnis

- [Br14] Brichzin, P.: Überwachung von Smartphone-Kosten. Ein Beispiel für die Aufgabenentwicklung mithilfe von Leitfragen. LOG IN 176/177, 82-87 (2014).
- [Br16] Braun, F.: Schülervorstellungen von Smartphones. Schr. Hausarbeit im Rahmen der Ersten Staatspr. f. d. Lehramt an Gymn./Gesamtschulen, Univ. Duisburg-Essen, 2016.
- [Di11] Diethelm, I.; Dörge, C.; Mesaros, A.; Dünnebier, M.: Die Didaktische Reduktion für den Informatikunterricht. In (Thomas, M. Hrsg.): Informatik in Bildung und Beruf. 14. GI-Fachtagung „Informatik und Schule – INFOS 2011“. Köllen, Bonn, S. 77-86, 2011.
- [DWZ12] Diethelm, I.; Wilken, H.; Zumbärgel, S.: An investigation of secondary school students' conceptions on how the Internet works. In (ACM ed.): Proc. of the 12th Koli Calling Int. Conf. on Computing Educ. Research. ACM, New York, pp. 67-73, 2012.
- [Mü11] Müller, D.: Fachdidaktisch begründete Auswahl von Informatiksystemen für den Unterrichtseinsatz. In (Thomas, M. Hrsg.): Informatik in Bildung und Beruf. 14. GI-Fachtagung „Informatik und Schule – INFOS 2011“. Köllen, Bonn, S. 167-176, 2011.
- [Pa05] Papastergiou, M.: Students' Mental Models of the Internet and Their Didactical Exploitation in Informatics Education. Educ. a. Inform. Techn. (10) 4, S. 341-360, 2005.

Das Konzept *Nature of Computer Science*

Yannick Schneider,¹ Andreas Mühling²

Abstract: Diese Arbeit präsentiert die Ergebnisse einer Delphi-Befragung, die unter Experten einen Konsens über das „Bild“, also die wissenschaftlichen Ziele und Methoden der Informatik, ermitteln soll. Die Experten wurden sowohl aus dem akademischen als auch industriellen Umfeld akquiriert. Ziel dieser und folgender Untersuchungen ist es, das Konstrukt „*Nature of Computer Science*“, d.h. angemessene Vorstellungen von Informatik, für Forschende und Lehrende in der Informatikdidaktik zugänglich zu machen.

Keywords: Informatikdidaktik; Nature of Computer Science; Informatikbild; Schülerperspektive

1 Einleitung und Stand der Forschung

Informatikunterricht kann viele Ziele verfolgen und auf viele Arten ausgestaltet sein. Während aktuell die Diskussion um „digitale Bildung“ der Informatik eine eher unterstützende Rolle im Kanon mit anderen Fächern zuweist, gibt es nach wie vor auch den Fachunterricht mit klarem Bezug zur Wissenschaft der Informatik. Ein Ziel sollte in einem solchen Unterrichtsszenario natürlich auch sein, diese Bezugswissenschaft kennenzulernen und ein adäquates Verständnis der Denkweisen und Methoden von Informatikern auszubilden. Befragungen mit Schülern und Studienanfängern zeigen, dass viele von ihnen nur unzureichende Vorstellungen von Informatik haben [He13, MW06]. Das mag einerseits an einem fehlenden Kontakt zu Informatikunterricht selbst liegen, aber auch an der uneinheitlichen Ausgestaltung des Fachunterrichts sowie an einem fehlenden Konsens unter den Experten hinsichtlich dem Bild der Wissenschaft Informatik [Ha93].

In diesem Artikel wird eine Erhebung nach der Delphi-Methode vorgestellt, die versucht, normativ unter Experten einen Konsens darüber zu ermitteln, was die Informatik als Wissenschaft auszeichnet.

In den Naturwissenschaften und der Mathematik werden derartige Forschungen seit Jahrzehnten unter den Schlagwörtern „Nature of Science“ und „Nature of Mathematics“ durchgeführt. Einen umfassenden Überblick über die bisherigen Arbeiten in den Naturwissenschaften und der Mathematik liefern [Le07, ODV02]. Im Bereich Informatik existieren bislang philosophische Arbeiten [Ha93] und einige deskriptive Studien aus dem amerikanischen Raum, deren Fokus auf den Einstellungen und dem Selbstkonzept von Schülern und Studierenden liegt [EDS12],[Wi03].

Im Rahmen des Projekts „Weltbilder der Informatik“ wurde eine qualitative Erhebung unter Studierenden durchgeführt [GI13]. „[D]as Bild der Informatik“ ist ein Bestandteil

¹ Christian-Albrechts-Universität zu Kiel, ysc@informatik.uni-kiel.de

² Christian-Albrechts-Universität zu Kiel, Arbeitsgruppe Didaktik der Informatik, am@informatik.uni-kiel.de

dieses Weltbildes. Teilweise wurden den Studierenden in der Weltbilder-Studie ähnliche Fragen gestellt wie den Teilnehmern in dieser Befragung, es geht im Gegensatz zu der hier präsentierten Studie allerdings nicht um einen Konsens unter Experten sondern um „die subjektiven bzw. kollektiven (alltagsweltlichen) Einschätzungen der Studierenden“. Eine vergleichbare Befragung [MW06] führten Maaß und Wiesner mit dem Fokus auf das Studienfach Informatik durch. In dieselbe Richtung gehen auch einige Studien amerikanischer Forscher, beispielsweise [He13].

2 Konzeption der Studie

Es wurde eine Befragung nach der Delphi-Methode organisiert. Hinweise zur Methodik finden sich u.a. in [Hä02]. Die Teilnehmer wurden sowohl aus dem akademischen Umfeld, über die GI-Fachgruppe „Didaktik der Informatik“, als auch aus der Industrie, über den lokalen Dachverband „Digitale Wirtschaft Schleswig-Holstein“ (DiWiSH), akquiriert um ein möglichst heterogenes Expertenpanel zusammenzustellen. Die Ergebnisse wurden qualitativ nach der Methode der Grounded Theory kategorisiert.

Die Eingangsfragen waren: „Wie definieren Sie für sich den Begriff ‚Informatik‘?“ und „Halten Sie ‚computer science‘ für eine angemessene Übersetzung von ‚Informatik‘? Falls nicht: Was stört Sie daran?“ Ziel dieser Fragen waren das Sammeln von persönlichen Bildern und eine Bewertung des englischen Begriffs „computer science“, der in der Informatik-Gemeinschaft seit jeher umstritten ist ([Co92], [Ja93]).

Die Methoden einer Wissenschaft und ihr Verhältnis zu anderen Wissenschaften sind ein elementarer Bestandteil ihres Selbstverständnisses und somit ihrer Natur [Ha93]. Die Experten wurden nach einzigartigen/typischen Methoden der Informatik befragt und gebeten, die Informatik in die vier Wissenschaftskategorien (nach [Ba08]) Natur-, Struktur-, Geistes- und Ingenieurwissenschaften einzuordnen (5-stufige Likert Skala).

Die zentralen fünf Fragen der ersten Runde waren Definitionen von oder Aussagen über Informatik, zu denen die Teilnehmer Stellung beziehen sollten. Als Quellen wurden die großen Dachverbände GI ³, ACM [De89] und CSTA [CS04] sowie die Studiengangsbeschreibung der RWTH Aachen ⁴ und die Definition eines klassischen Informatiklehrbuches [Br98] verwendet.

Die beiden letzten Fragen befassen sich mit „häufigen Missverständnis[en] in Bezug zur Informatik (von Schülern, Studierenden, Auszubildenden, Kunden oder Laien)“ und Aspekten, in denen Informatik häufig unterschätzt wird.

³ Was ist Informatik? Unser Positionspapier. <https://www.gi.de/fileadmin/redaktion/Download/was-ist-informatik-lang.pdf>, 2006

⁴ Course of Study Description: Computer Science B.Sc. <http://www.rwthachen.de/cms/root/Studium/Vor-dem-Studium/Studiengaenge/Liste-Aktuelle-Studiengaenge/Studiengangbeschreibung/bnzs/Informatik-B-Sc/>

3 Ergebnisse und Ausblick

An der ersten Runde der Befragung nahmen 14 Teilnehmer teil. Sechs davon kamen aus dem akademischen, acht aus dem industriellen Umfeld. In der zweiten Runde war das Panel auf acht Teilnehmer, je vier mit akademischen und industriellem Hintergrund, geschrumpft. Daher und aufgrund der Tatsache, dass unter den verbleibenden Teilnehmern weitgehende Übereinstimmung herrschte, wurde auf eine dritte Runde verzichtet.

Konsens herrschte über den Begriff „computer science“, den fast alle (13 von 14) Teilnehmer als unangemessen bewerten. Große Einigkeit gibt es außerdem hinsichtlich der Eingangsfrage. Neun der 14 Teilnehmer definierten Informatik für sich selbst sinngemäß als „Wissenschaft der automatischen Informationsverarbeitung“. Die dritte Frage, nach den typischen oder einzigartigen Methoden der Informatik, ließ sich nicht zu einem Konsens führen. Die meistgenannten typischen, aber nicht notwendigerweise einzigartigen, Methoden waren: Modellierung (6×), Verwendung digitaler Daten und Implementierung (4×), algorithmisches Problemlösen, Abstraktion, Konstruktion und Dekonstruktion sowie die Verwendung einer universellen Maschine (je 2×).

Die Einordnung in die Wissenschaftskategorien liefert ein gemischtes Bild. Die größte Einigkeit besteht in der Ablehnung bei Geisteswissenschaft und Zustimmung bei Ingenieurwissenschaft. Strukturwissenschaft bewegt sich um die Mitte der Likert Skala, während Naturwissenschaft eine eher bimodale Verteilung zeigt (6 mal Ablehnung, 5 mal Zustimmung). Die beiden Teilnehmer, die hier „eindeutig nein“ gewählt hatten, stimmten in ihren Begründungen überein: die Informatik beschäftige sich nicht mit natürlichen, sondern mit menschengemachten Phänomenen. Der Teilnehmer, der „eindeutig ja“ gewählt hatte ist der Ansicht, dass die Gesellschaft inzwischen so sehr von Informatiksystemen durchdrungen ist, dass diese quasi-natürliche Phänomene bildeten und auch als solche erforscht würden.

Die Auswertung der Kommentare zu den gegebenen Definitionen drehte sich hauptsächlich um eine Kritik an der Über- bzw. Unterrepräsentation von Themen. Insgesamt fünf Teilnehmer monierten, die Definitionen von ACM und CSTA seien zu stark auf Algorithmik und Automatisierung fokussiert. Da diese beiden Definitionen als einzige dem englischsprachigen Kulturkreis entstammen, könnte auch dies wieder ein Indiz darauf sein, dass mit dem Begriff „computer science“ eine andere Sichtweise auf die Disziplin Informatik verknüpft ist. Fünf Teilnehmer rekurrten auf den Themenkomplex „Informatik und Gesellschaft“, es wurde sowohl an den Definitionen von Broy und der GI dessen Fehlen bemängelt als auch an den Definitionen von CSTA und RWTH deren Nennung honoriert.

Das Verhältnis von Informatik und Programmierung – in Vorarbeiten teilweise gleichgesetzt [Wi03] – erweist sich als nicht eindeutig: das Panel erkennt zwar das Implementieren oder Programmieren als typische Methode an, sieben Teilnehmer gaben aber als häufigstes Missverständnis die Gleichsetzung von Informatik und Programmierung an.

Zusammengefasst kann man als Konsens des Panels festhalten: Die Informatik begreift sich selbst als Bindeglied zwischen Ingenieurs- und Strukturwissenschaften. Sie befasst sich mit dem Untersuchungsobjekt „Information“ mit den eng damit verknüpften Metho-

den der Abstraktion und Modellierung. Eine Verzahnung von Forschung und Entwicklung, verknüpft mit der Methode des Implementierens, ist typisch. Zu den Aufgaben eines Informatikers gehört auch die Beschäftigung mit den sozialen Auswirkungen und der Interaktion von Mensch und Computer.

Literaturverzeichnis

- [Ba08] Balzert, Helmut: *Wissenschaftliches Arbeiten*. Herdecke, 2008.
- [Br98] Broy, Manfred: *Informatik. Eine grundlegende Einführung*. Berlin, 1998.
- [Co92] Coy, Wolfgang: Für eine Theorie der Informatik! In (Coy, Wolfgang et al., Hrsg.): *Sichtweisen der Informatik*. Braunschweig, 1992.
- [CS04] CSTA, Computer Science Teachers Association: *Model Curriculum for K-12 Computer Science*. 2004.
- [De89] Denning, Peter J. et al.: Computing as a discipline. *Communications of the ACM*, 32(1):9–23, 1989.
- [EDS12] Elliott Tew, Allison; Dorn, Brian; Schneider, Oliver: Toward a Validated Computing Attitudes Survey. In: *Proceedings of the Ninth Annual International Conference on International Computing Education Research*. ICER '12, New York, S. 135–142, 2012.
- [GI13] GI, Gesellschaft für Informatik, Hrsg. Sonderheft: *Weltbilder in der Informatik: Sichtweisen auf Profession, Studium, Genderaspekte und Verantwortung*, Jgg. 36(3) in *Informatik-Spektrum*. 2013.
- [Ha93] Hartmanis, Juris: Some observations about the nature of computer science. *Lecture Notes in Computer Science*, 761:1–12, 1993.
- [Hä02] Häder, Michael: *Delphi-Befragungen. Ein Arbeitsbuch*. Wiesbaden, 2002.
- [He13] Hewner, Michael: Undergraduate Conceptions of the Field of Computer Science. In (Simon, Beth; Clear, Alison; Cutts, Quintin I., Hrsg.): *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, La Jolla, USA, 12–14 August 2013. New York, S. 107–114, 2013.
- [Ja93] Janich, Peter: Zur Konstitution der Informatik als Wissenschaft. In (Scheffe, Peter; Hastedt, Heiner; Dittrich, Yvonne; Keil, Geert, Hrsg.): *Informatik und Philosophie*. Mannheim, S. 53–68, 1993.
- [Le07] Lederman, Norman G.: Nature of Science: Past, Present and Future. In (Abell, Sandra K.; Lederman, Norman G., Hrsg.): *Handbook of research on science education*. Mahwah, 2007.
- [MW06] Maaß, Susanne; Wiesner, Heike: Programmieren, Mathe und ein bisschen Hardware ... Wen lockt dies Bild der Informatik? *Informatik-Spektrum*, 29(2):125–132, 2006.
- [ODV02] Op't Eynde, Peter; De Corte, Erik; Verschaffel, Lieven: Framing Students' Mathematics-Related Beliefs. A Quest for Conceptual Clarity and a Comprehensive Categorization. In (Leder, Gilah C.; Pehkonen, Erkki; Törner, Günter, Hrsg.): *Beliefs: A Hidden Variable in Mathematics Education?* Dordrecht, S. 13–37, 2002.
- [Wi03] Wiebe, Eric; Williams, Laurie; Yang, Kai; Miller, Carol: Computer science attitude survey. *computer science*, 14(25):0–86, 2003.

Berufswahl Informatiklehrkraft

Dorothee Müller¹

Abstract: Der Mangel an qualifizierten Informatiklehrkräften ist ein seit Jahren bestehendes, deutschlandweites Problem. Die aktuellen Studierendenzahlen und das hohe Durchschnittsalter der aktiven Informatiklehrkräfte machen eine Verschärfung des Problems innerhalb der nächsten Jahre voraussehbar. In einer empirischen qualitativen Untersuchung wurden Faktoren identifiziert, die die Berufswahl Informatiklehrkraft positiv oder negativ beeinflussen. In diesem Beitrag werden die Ausgangssituation, der wissenschaftliche Kontext, die Methoden und die Ergebnisse der Untersuchung dargelegt. Die Ergebnisse zeigen, dass dem Bild der Informatik, dem Bild des Informatikunterrichts und dem Informatikselbstkonzept eine herausragende Rolle bei der Berufswahl Informatiklehrkraft zukommen. Diese Faktoren werden entscheidend von Informatikunterricht bestimmt.

Keywords: Lehrerbildung Informatik, Informatiklehrermangel, Berufswahlmotivation Informatiklehrkraft, Bild der Informatik, Informatikselbstkonzept, Informatikunterricht

1 Ausgangssituation Informatiklehrermangel

Seit Jahren ist der Mangel an Informatiklehrkräften bekannt [FMW06] und wird fachdidaktisch und politisch diskutiert, ohne dass die Ursachen empirisch erforscht oder geeignete bildungspolitische Maßnahmen ergriffen wurden. Es ist prognostizierbar, dass sich der Mangel in den kommenden Jahren noch deutlich vergrößern wird: Mehr als 56 % der Informatiklehrkräfte in Nordrhein-Westfalen² sind älter als 50 Jahre und werden in den nächsten 15 Jahren aus dem Schuldienst ausscheiden. Aufgrund der Anzahl von Studierenden mit dem Berufsziel Informatiklehrkraft kann in den nächsten Jahren durchschnittlich mit 55 Bewerbern für Informatiklehrkraftstellen in Nordrhein-Westfalen gerechnet werden. Damit können nur 56 % des Neubedarfs gedeckt werden [K115, S. 19]). Neben dem hohen Durchschnittsalter der Informatiklehrkräfte ist der sehr geringe Anteil von 25,7 % an weiblichen Lehrkräften im Fach Informatik [MS15, S. 13, 57 f.] auffällig.

Bereits jetzt führt der Informatiklehrermangel dazu, dass Informatikunterricht nicht angeboten wird, ausfällt oder fachfremd, d. h. von Lehrkräften ohne Lehrbefähigung für das Fach Informatik, unterrichtet wird (s. Tab. 1). Der hohe Anteil an fachfremdem Informatikunterricht ist für die informatische Bildung besonders bedrohlicher, weil die Lehrkraft als der entscheidende Faktor für den Erfolg schulischen Lernens erkannt wurde [Ha09]. Und empirische Studien bestätigen »signifikante Zusammenhänge zwischen der fachbezogenen Lehrbefähigung und den erreichten Schülerkompetenzen« [Pa13, S. 283].

¹ Bergische Universität Wuppertal, Didaktik der Informatik, 42119 Wuppertal, dmueller@uni-wuppertal.de

² Für die in diesem Beitrag dargestellte empirische Untersuchung wurden Befragungen in Nordrhein-Westfalen durchgeführt, und es wurden entsprechend vor allem statistische Daten zum Schulwesen in Nordrhein-Westfalen herangezogen. Die Ausgangssituation ist jedoch in anderen Bundesländern ähnlich [K115, S. 3].

Schulform	erteilte Stunden	ohne Lehrbefähigung
Hauptschule	2.436	84,7 %
Realschule	6.640	54,4 %
Gesamtschule	2.094	74,1 %
Gymnasium	4.280	36,1 %

Tab. 1: Informatikunterricht 2015/16 in NRW: erteilte Stunden und Anteil des Unterrichts ohne Lehrbefähigung [PM16, S. 3]

2 Forschungsstand Berufswahl Informatiklehrkraft

Der bundesweite Mangel an Informatiklehrkräften lenkt den Blick der Fachdidaktik Informatik stärker auf das Studium mit dem Ziel Lehramt Informatik und die entsprechenden Berufswahlmotive. So schlagen Informatikdidaktiker im Zusammenhang mit der Lehrerbildung Informatik vor, Vorstellungen zu erforschen, die zu einem Studium Lehramt Informatik geführt haben oder die es verhindert haben [Di10, S. 59]. Auf bereits durchgeführte informatikfachdidaktische Untersuchungen zur Berufswahl Informatiklehrkraft kann nicht zurückgegriffen werden. Lediglich Teilaspekte wurden bisher empirisch untersucht. So fragt Humbert sechzehn aufgrund ihres fachdidaktischen Engagements ausgewählte Informatiklehrkräfte nach Gründen für Ihre Hinwendung zur Wissenschaft Informatik [Hu03, S. 88]. Auch die Untersuchung von Knobelsdorf zur Computernutzung als einen Weg in die Informatik [Kn11] thematisiert nicht speziell den Weg ins Lehramt Informatik, obwohl neben anderen Studierenden auch Studierende mit dem Ziel Lehramt Informatik befragt wurden.

Die Berufswahl Informatiklehrkraft kann als zwei – eventuell gleichzeitig getroffene – Teilentscheidungen aufgefasst werden: als Berufswahl Lehrkraft und als Berufs-/Studienwahl Informatik. Für beide getrennt betrachteten Einzelentscheidungen existieren Forschungsbeiträge, die eine Grundlage für die hier dargestellte Untersuchung liefern.

Zur *Berufs-/Studienwahl Informatik* werden immer wieder Forschungsprojekte und Fördermaßnahmen (vgl. z. B.: [Kl13, GH13]) durchgeführt. Das Bild der Informatik, das Informatikselbstkonzept und die eventuelle Enkulturation in die Informatik beeinflussen die Berufs-/Studienwahl Informatik entscheidend. Die drei Faktoren werden durch den Schulunterricht beeinflusst. Zwischen der *Leistungskurswahl* und der späteren Studien- oder/und Berufswahl besteht eine direkte, empirisch nachgewiesene Beziehung. Im Vergleich mit anderen MINT-Fächern³ wird deutlich, dass relativ wenige Schülerinnen und Schüler in Nordrhein-Westfalen an Leistungskursen Informatik teilnehmen: Nur 1.330 Schülerinnen und Schüler besuchen einen Leistungskurs Informatik, im Vergleich etwa fünfmal so viele (6.557) einen Leistungskurs Chemie und fast dreißig mal so viele einen Leistungskurs Biologie (39.102). Die Leistungskurswahl resultiert wiederum hauptsächlich aus dem in der Sekundarstufe I geweckten Interesse.

³ MINT ist ein Akronym für Mathematik, Informatik, Naturwissenschaften und Technik.

Auch die *Berufswahl von Lehrkräften* ist ein Forschungsgegenstand, zu dem immer wieder neue, aktuelle Untersuchungen durchgeführt werden. Vor allem die Fragen nach Rekrutierungsmustern und Berufswahlmotiven stehen im Mittelpunkt der Untersuchungen. Bei der Frage, *wer* den Beruf Lehrkraft wählt, werden meist Merkmale wie Geschlecht, Herkunft, Schichtzugehörigkeit, Milieuzugehörigkeit, Persönlichkeit und Leistungsmerkmale quantitativ erfasst (Übersicht über den Forschungsstand in [Ro14b]). Zur Beantwortung der Frage, *warum* der Beruf Lehrkraft ergriffen wird, werden die Interessen, Motive und Überzeugungen, die zu der Berufsentscheidung führen, quantitativ erhoben (Übersicht über den Forschungsstand in [Ro14a]). Als wichtig für die Berufswahl Lehrer im Allgemeinen wie auch für die Fächerwahl hat sich der Einfluss der eigenen Schulzeit erwiesen. Zukünftige Lehrkräfte greifen bei der Wahl der Studienfächer oft auf ihre »Lieblingsfächer in der Schule« [U104] zurück. Diese Fächerwahl wird durch die Leistungskurswahl vorbereitet und angekündigt [K107].

Trotz der zahlreichen Forschungsbeiträge zur Berufswahl von Lehrkräften wird in diesem Zusammenhang von Forschungsdesideraten gesprochen. Dies betrifft vor allem zwei Aspekte dieser Forschung: die fehlende Differenzierung nach Schulfächern und die Forschungsperspektive einschließlich der damit verknüpften Methodenwahl. Vergleichende Studien [Ka12] zeigen jedoch, dass sich Lehramtsstudierende verschiedener Fachkombinationen bezüglich Interessenstrukturen, kognitiver Leistungsfähigkeit, Studienzufriedenheit und Leistung deutlich voneinander unterscheiden und Differenzierung nach Fächern somit notwendig ist. Das zweite Desiderat im Zusammenhang mit der Forschung zur Berufswahl von Lehrkräften betrifft die Perspektive und Methoden dieser Forschung. Die empirische Forschung zur Berufswahl von Lehrkräften ist meistens quantitativ orientiert. Es fehlt die qualitative Forschung mit einer biografischen Perspektive auf den zum Lehrerberuf führenden Bildungsprozess. Diese Forschung könnte weitere, eventuell unerwartete Einflussfaktoren ermitteln.

3 Forschungsziel und Methodendiskussion

3.1 Forschungsziel

Die hier dargestellte Untersuchung [Mü17] hat das Ziel, die beschriebenen Forschungsdesiderata zu verringern: Die Berufswahl von Informatiklehrkräften wurde aus der individuellen, biografischen Perspektive der Beteiligten erforscht. Daten zu spezifischen Einflussfaktoren der Berufswahl speziell dieser Gruppe von Lehrkräften wurden in offenen qualitativen Verfahren erhoben und ausgewertet. Dabei wurden Faktoren identifiziert, die die Berufswahl Informatiklehrkraft positiv oder negativ beeinflussen.

3.2 Qualitative empirische Forschung nach der Grounded Theory

Da kaum Forschungsergebnisse und keine empirisch fundierten Theorien zur Berufswahl von Informatiklehrkräften existieren, mussten im Forschungsprozess Theorien generiert

werden. Als methodologisches Rahmenkonzept wurde die Grounded Theory gewählt. Diese eignet sich für die Theoriegenerierung auf der Grundlage von qualitativen empirischen Daten in einem noch relativ unerforschten Feld. ngsprozess.

Stichprobe: Die Stichprobe wurde mit dem für die Grounded Theory typischen Verfahren des Theoretical Samplings zusammengestellt. Die Gruppe der Befragten wurde nicht komplett vor Beginn der Datenerhebung zusammengestellt, sondern es wurde bei den beiden parallel verlaufenden Prozessen der Datenerhebung und der Datenauswertung vor der Aufnahme neuer Fälle geprüft, ob durch weitere Datenerhebungen neuen Erkenntnisse zu erwarten waren. Als Grundgesamtheit der zu befragenden Personen wurden Studierende mit dem Ziel Lehramt Informatik nach oder bei der ersten besuchten Veranstaltung zur Didaktik der Informatik gewählt. Für diese Entscheidung sprach, dass sich mit der Teilnahme an dieser Veranstaltung auch in dem polyvalenten Bachelor-Studium die Entscheidung für den Beruf Informatiklehrkraft dokumentiert und zugleich der zeitliche Abstand zu der Entscheidung relativ gering ist. Dass der Kreis der Befragten sich ausschließlich aus Studierenden zusammensetzt, die sich für das Berufsziel Informatiklehrkraft entschieden haben, wirft die methodenkritische Frage auf, ob dadurch größtenteils Faktoren identifiziert werden, die die Berufswahl Informatiklehrkraft befördern, oder ob auch Faktoren ermittelt werden können, die die Berufswahl behindern.

Datenerhebung: Die Datenerhebung erfolgte als eine Kette von aufeinander aufbauenden Auswahlentscheidungen, wobei die Auswahl der Erhebungsinstrumente und die Erhebung der Daten sukzessive erfolgte. Drei Erhebungsformen wurden in chronologischer Folge genutzt: das teilnarrative Leitfadeninterview, die Gruppendiskussion zum Thema Berufswahlbiografie und das speziell für diese Untersuchung entwickelte Erhebungsinstrument der schriftlichen Berufswahlbiografie. Die schriftliche Berufswahlbiografie stellte sich als besonders geeignet heraus, da sie durch die schriftliche Form die Fokussierung auf das Thema förderte und zugleich die Kontrolle der Befragten über die Preisgabe der eigenen biografischen Daten garantierte.

Datenauswertung: Die erhobenen Daten wurden in einer Methodentriangulation aus den Verfahren der Grounded Theory und der dokumentarischen Methode ausgewertet. Angewandt wurde die Kodierungsmethode der Grounded Theory, ein kategorisierendes Verfahren mit dem Ziel der Gewinnung von Kategorien und die Generierung von Theorien. Ergänzt wurde dieses Verfahren für die interpretierende Arbeit am Datenmaterial durch die dokumentarische Methode nach Ralf Bohnsack.

4 Durchführung der Befragung

Befragt wurden insgesamt 45 Studierende mit dem Studienziel Lehramt Informatik. Davon waren zehn weiblich. Es wurden neun mündliche Berufswahlinterviews und 34 schriftliche Berufswahlbiografien erhoben, sodass von 43 Befragten Berufswahlinterviews oder -biografien vorliegen. Zwei der elf Teilnehmenden der Gruppendiskussionen nahmen nur an diesen und nicht ebenfalls an mündlichen oder schriftlichen Interviews teil. Aussagen dieser Teilnehmenden wurden in der Auswertung berücksichtigt.

Die Reihe der Datenerhebungen wurde mit neun aufgezeichneten teilnarrativen Leitfadeninterviews eröffnet. Ausgegangen wurde im *Leitfadeninterview* von einer offen gehaltenen Frage zur Berufswahlbiografie: »Ich möchte Sie zu Beginn ganz einfach bitten, mir zu erzählen, wie es kommt, dass Sie heute hier als Studierende(r) mit dem Berufsziel Informatiklehrer(in) sitzen. Sie können so weit ausholen, wie sie wollen. Mich interessiert, wie eins zum anderen gekommen ist.« Die Interviews dauerten jeweils etwa 30 Minuten. Falls die Befragten folgende Themenkomplexe nicht oder kaum thematisierten, wurden vorbereitete Leitfragen zur eigenen Computerbiografie, zum Einfluss von Freunden und Familie und zum antizipierten Berufsbild Informatiklehrkraft ergänzt.

Es wurden zwei *Gruppendiskussionen* durchgeführt. Mit der ersten Diskussion zum Thema »Mädchen im Informatikunterricht« sollte vor allem die Erhebungs- und Auswertungsform erprobt werden. Die zweite Gruppendiskussion zum Thema »Warum werde ich Informatiklehrkraft?« fokussiert dagegen die Berufswahlmotive der Teilnehmenden. Das Erhebungsinstrument der Gruppendiskussionen stellte sich für diesen Forschungszweck als weniger ergiebig als erwartet heraus und floss nur zu geringem Teil in die Auswertung ein.

In der *schriftlichen Berufswahlbiografie Informatiklehrkraft* wurde an die Befragten eine schriftliche Schreibaufforderung gegeben. Diese bestand aus der eigentlichen Aufforderung und kurzen »Locktexten«, die den Erzählraum beispielhaft inhaltlich und formal aufschlossen und wichtige Themen in den gedanklichen Fokus der Befragten brachten, ohne Aussagen zu den Themen zu erzwingen.

5 Ergebnisse

Die Datenauswertung nach der Kodierungsmethode der Grounded Theory führte zu den folgenden Haupt- und untergeordneten Subkategorien:

- Irrwege und Wege zur Berufswahl Informatiklehrkraft
 - Informatiklehrkraft – zwei Berufsentscheidungen
 - Vom Abitur direkt zum Studium Informatiklehrkraft
 - Über Irrwege zur Berufswahl
 - Faktor Zufall
- Bild der Informatik
 - Philosophie des Faches / epistemologische Überzeugungen
 - Bild des Informatikers und des Berufshabitus Informatiklehrkraft
- Informatikselbstkonzept
 - Eigene Schulerfahrungen im Informatikunterricht
 - Erfahrungen mit Informatiksystemen
 - Familie und Peer-Gruppe
 - Geschlecht und Informatik

Ziel der folgenden Darstellung ist es, die vielfältigen Aspekte der Forschungsergebnisse strukturiert und in dem Zusammenhang der »Handlung in dieser Geschichte«, also des zentralen Themas in Sinne Corbins [Co06, S. 74], vorzustellen. Als diese Schlüsselkategorie kristallisierte sich bei der Auswertung »Irrwege und Wege zur Berufswahl Infor-

matiklehrkraft« heraus. Die Kategorien sind in der Ergebnisdarstellung nicht vollständig trennbar, da sie sich in vielen Aspekten wechselseitig bedingen. Beispielzitate aus dem erhobenen Datenmaterial belegen und illustrieren die Ergebnisse.

Die Ergebnisse der Untersuchung zeigen, dass der Berufswahlprozess von angehenden Informatiklehrkräften häufig mit *Irrwegen* in Form von Studien- bzw. Berufszielwechsell verbunden ist. Die Entscheidung für den Beruf Informatiklehrkraft wurde meist als eine *doppelte Berufsentscheidung*, als die Entscheidung für den Beruf Lehrkraft und als die Entscheidung für das Fach Informatik, begriffen und auch formuliert.

»Ja, wobei das ein Unterschied ist, Lehrer oder Informatiklehrer. Informatiklehrer wollte ich eben noch nicht lange werden, das hat sich eben so ergeben.«

»Die Wahl Informatik und Lehrer sind voneinander unabhängig getroffen worden.«

Diese doppelten Berufsentscheidungen werden meist auch zeitlich getrennt getroffen und führen zu Umwegen im Berufsweg. Von der in der Berufsentscheidung Informatiklehrkraft enthaltenen doppelten Entscheidung für Lehramt und Informatik ausgehend können die Wege zur Berufsentscheidung Informatiklehrkraft in drei Arten eingeteilt werden: den Weg vom Abitur direkt zum Studium Informatiklehrkraft, den Weg über ein Lehramtsstudium ohne Informatik und den Weg über die Informatik ohne das Ziel Lehramt. Nur ein knappes Drittel der Befragten traf nach dem Schulabschluss als erste Berufswahl die Entscheidung für den Beruf Informatiklehrkraft. Stattdessen wurde die Entscheidung meistens erst nach oder während einer Berufsausbildung oder während eines anderen Studiums getroffen. Dies kann als Grund für das von der Statistik ausgewiesene (vgl. z. B. [St14, S. 164]) vergleichsweise hohe Durchschnittsalter bei gleichzeitig geringer Fachstudienzeit bei Absolventen einer Lehramtsprüfung mit dem Fach Informatik gesehen werden.

Der Zusammenhang zwischen in der eigenen Schulzeit entwickeltem Fachinteresse, der Leistungskurswahl und der Studienfachwahl bestätigt sich für diejenigen Befragten, die den *»Weg vom Abitur direkt zum Studium Informatiklehrkraft«* gingen. Häufig wird der Informatikunterricht der eigenen Schulzeit positiv erinnert, und diese Erfahrung fließt explizit in die Entscheidung ein. Sie hatten zur Schulzeit ein positives Bild der Informatik und ein hohes Informatikselbstkonzept. Der Informatiklehrkraft ihrer Schulzeit bescheinigen sie oft eine berufliche Vorbildfunktion.

»Und auch in der Schule haben Mathe und Info am meisten Spaß gemacht und deswegen studiere ich es wahrscheinlich.«

»Die Situation von meinem [Informatik-]Lehrer hat mich im Endeffekt davon überzeugt, dass auch Lehrer ihren Spaß am Unterricht und am Unterrichten haben können.«

Allerdings hatten die meisten der befragten angehenden Informatiklehrkräfte selbst keinen Informatikunterricht oder erinnern diesen negativ. Der Weg zum Studium Informatiklehrkraft führt bei diesen Befragten häufig über den Umweg von zunächst anderen Studienentscheidungen, meistens über ein Lehramtsstudium mit anderen Fächern oder ein Informatikstudium. Die Informatikstudierenden haben typischerweise zum Zeitpunkt ihrer ersten Studienwahl ein positives Bild der Informatik und ein hohes Informatikselbstkonzept, aber

kein positives Berufsbild Informatiklehrkraft. Ihr Wechsel von einem Informatikstudium zum Studium mit dem Berufsziel Informatiklehrkraft wird in der Regel durch den Wunsch nach einer stärkeren sozialen Komponente im späteren Berufsalltag ausgelöst.

»Dies ist der erste Grund, weshalb ich mich für ein Lehramtsstudium entschied. Mir fehlte einfach die soziale Komponente meiner (bis dato) Tätigkeit [Programmieren d. V.].«

Oder ihr Berufsbild von Informatiklehrenden erfährt eine entscheidende Änderung.

»Später im Studium haben mich dann Professoren, die Informatik erteilten, davon überzeugt, dass es Spaß macht diesen Stoff zu vermitteln. Dieser Gedanke kam bei mir nicht durch ein Gespräch zustande, sondern durch die Teilnahme an Vorlesungen und den dazugehörigen Übungen.«

Bei den Lehramtsstudierenden mit anderen Fächern, die häufig zunächst ein niedriges Informatikselbstkonzept und/oder ein negatives Bild der Informatik hatten, kann es zu einer Umorientierung hin zum Studienziel Informatiklehrkraft kommen, wenn diese Vorstellungen sich während des ersten Studiums – z. B. durch den Besuch von universitären Lehrveranstaltungen zu informatischen Inhalten – ändern.

Die letztliche Entscheidung für den Beruf Informatiklehrkraft wird von denjenigen, die ihr Studienziel wechselten, mit Recht als durch Zufälle bestimmt empfunden. Immer wieder wird im Zusammenhang mit dem Berufszielwechsel von den Befragten explizit der Faktor »Zufall« thematisiert. So beginnen viele erhobene schriftliche Berufswahlbiografien mit Worten wie:

*»Ich bin also eher zufällig zur Informatik gekommen.«
 »Zum Berufswunsch ›Informatiklehrer‹ brachten mich eine Reihe von Zufällen.«
 »Ich werde eher zufällig Informatiklehrer.«*

Die dargestellten Ergebnisse zeigen, dass das Bild der Informatik und das damit zusammenhängende Informatikselbstkonzept entscheidend für die Berufswahl Informatiklehrkraft sind. Beide erhalten im Allgemeinen schon vor dem ersten Informatikunterricht eine vorläufige Prägung. Das stereotype, durch die Medien vermittelte und in der Gesellschaft vorherrschende Informatikbild wirkt sich dabei negativ aus und verhindert eventuell die Wahl des Faches Informatik. Ein Studierender mit dem Ziel Lehramt, der erst während des Studiums von einem anderen Fach zur Informatik wechselte, berichtet zum Beispiel:

»Ich hatte während meiner eigenen Schulzeit fast keinerlei Berührungen mit Informatik [...] Die allgemeinen Vorurteile, wie ›Informatik gleich Programmieren‹ oder ›Informatik ist für Computer-Freaks‹, hielten mich ab, dem Thema / der Wissenschaft die nötige Aufmerksamkeit zu schenken.«

Jedoch haben einige Befragte auch von einem bereits vor dem ersten Informatikunterricht bestehenden positiven Informatikbild berichtet. Dieses war meist mit Vorbildern in der Familie, die einen Beruf im Bereich der Informatik ausübten, verbunden.

Ein neutrales oder positives Bild der Informatik kann durch die Schule geändert werden – sogar ohne dass die Schülerin oder der Schüler jemals am Informatikunterricht teilnimmt. Ein Befragter berichtet, dass er ursprünglich durchaus Interesse an Informatik und am Informatikunterricht hatte, ihm jedoch in der schulischen Beratung vom Wahlpflichtfach Informatik abgeraten wurde, mit der Begründung, es gäbe »in Informatik nichts ›Wichtiges‹ zu lernen«.

Auch Informatikunterricht von fachlich oder/und fachdidaktisch unqualifizierten Lehrkräften kann sich negativ auf das Bild der Informatik oder/und das Informatikselbstkonzept auswirken. Auf die Frage, ob er Informatikunterricht in der eigenen Schulzeit hatte, antwortete ein Lehramtsstudierender, der erst während des Studiums von einem anderen Fach zur Informatik wechselte:

»Ja. Aber den habe ich nachher abgewählt, weil ich fand den aber, glaube ich, eher Kacke und eher uninteressant. [...] Also dass mir Programmieren Spaß macht, habe ich erst in der Uni gelernt. Weil in der Schule war das so – lag natürlich auch am Lehrer – ich hatte einen Lehrer, der hat gesagt, ihr habt frei, wenn er sein Buch vergessen hatte. [...] Also er hatte selber gar keine Ahnung, und konnte das Interesse oder den Spaß an dem Fach überhaupt nicht vermitteln.«

Das Bild der Informatik dieses Befragten änderte sich erst an der Universität, wo er feststellt, dass sein in der Schulzeit gewonnenes Informatikbild falsch war. Auch das Informatikselbstkonzept kann unter misslungenem Informatikunterricht leiden und eine weitere Beschäftigung mit Informatik verhindern:

»Informatik kam für mich zunächst nicht in Frage. Als Schüler hatte ich als Differenzierungskurs Mathematik/Informatik und da ich dort im Bereich Programmierung hinter denjenigen Mitschülern, die sowieso schon programmieren konnten, deutlich im Nachteil war, habe ich es nicht in Betracht gezogen, Informatik als Fach zu studieren.«

Dieser Befragte entwickelt in diesem Informatikunterricht ein niedriges, und wie sich im späteren Studium zeigt, nicht realistisches Informatikselbstkonzept. Erst im Studium kommt es zu einem erneuten Kontakt mit informatischen Themen, wobei sich sein Informatikselbstkonzept ebenso wie sein Bild der Informatik ändern und er zum Studienfach Informatik wechselt.

»Nachdem ich feststellte, dass mir Informatik liegt, [...] was vielleicht auch daran lag, dass [...] die Programmierung einfach ist, wenn man sie von Grund auf lernt [...] habe ich mich [...] dann endgültig für Lehramt Mathematik/Informatik entschieden.«

6 Fazit und Ausblick

Die Berufswahl Informatiklehrkraft wird vom informatischen Selbstkonzept der Wählenden, ihrem Bild der Informatik und ihrem Bild des Berufs Informatiklehrkraft bestimmt.

Auf alle diese Faktoren hat das Erleben oder auch Fehlen des eigenen Informatikunterrichts der Befragten entscheidenden Einfluss. Die Ergebnisse zeigen, dass die in der Fachliteratur als typisch beschriebenen Faktoren der Fächerwahl bei Lehramtsstudierenden für Informatiklehrkräfte nicht die Norm sind. Typischerweise studieren angehende Lehrkräfte die Fächer, für die ihr Interesse in der Sekundarstufe I geweckt wurde und die sie anschließend in der Sekundarstufe II oft als Leistungskurse oder zumindest als Grundkurse gewählt haben.

In der Gruppe der befragten angehenden Informatiklehrkräfte ist die Wertung des als Schülerin oder Schüler erlebten Informatikunterrichts weitgehend dichotom. Der größere Teil der Befragten hatte keinen Informatikunterricht oder hat eher negative Erfahrungen damit gemacht. Es wurde ein niedriges Informatikselbstkonzept aufgebaut und/oder ein negatives Bild der Informatik. Ein Studium mit dem Ziel Lehramt Informatik wird zunächst nicht in Betracht gezogen. Diejenigen, die sich zunächst für ein Fachstudium Informatik oder einen informatiknahen Beruf entschieden, waren häufig schon vor dem ersten möglichen Informatikunterricht von der Informatik begeistert. Sie wählten diesen eventuell nicht, weil sie davon keinen wesentlichen Lernzuwachs erwarteten. Erst positiv erlebte Informatiklehre an der Universität und/oder der Wunsch nach einer sozialen Komponente im Beruf führte zu einem Wechsel der Berufsziels zur Informatiklehrkraft. Es gibt jedoch auch Befragte, die in ihren Berufswahlbiografien mit Begeisterung von dem Informatikunterricht ihrer Schulzeit und den Informatiklehrkräften berichten. Die Entscheidung für den Beruf Informatiklehrkraft wurde bei diesen Befragten oft als erste Berufsentscheidung nach dem Abitur betroffen. In diesen Fällen wird oft explizit auf die Vorbildfunktion ihrer Informatiklehrkraft und deren Einfluss auf ihre Berufswahl Informatiklehrkraft hingewiesen.

Ein fachlich und fachdidaktisch guter Informatikunterricht ist der förderlichste Faktor für die Berufswahl Informatiklehrkraft. Ein diesen Anforderungen entsprechender Informatikunterricht für alle Schülerinnen und Schüler kann dauerhaft dem Informatiklehrermangel entgegenwirken.

Keine fundierten Ergebnisse konnten anhand der Befragungen zu dem Thema Geschlecht und Informatik ermittelt werden. Der niedrige Anteil von weiblichen Informatiklehrkräften und die allgemein übliche Zuordnung der Informatik zur männlichen Domäne lässt eine spezielle Untersuchung zu diesen Thema wichtig erscheinen.

Literaturverzeichnis

- [Co06] Corbin, Juliet: *Grounded Theory*. Verlag Barbara Budrich, Opladen und Farmington Hills, S. 70–75, 2006.
- [Di10] Diethelm, Ira; Hellmig, Lutz; Friedrich, Steffen; Breier, Norbert; Brinda, Torsten: *Lehrerbildung Informatik – Was ist zu tun?* In (Diethelm, Ira; Dörge, Christina; Hildebrandt, Claudia; Schulte, Carsten, Hrsg.): *DDI*. Jgg. 168 in *LNI*. GI, S. 57–65, 2010.
- [FMW06] Frein, Thomas; Möller, Gerd; Wilpricht, Michael: *Fachspezifischer Lehrermangel am Gymnasium: Mythos oder Wahrheit?* *Schulverwaltung NRW*, (1):29, Januar 2006.
- [GH13] Götz, Christian; Hubwieser, Peter: *Belebt die Einführung des Schulfachs Informatik die Nachfrage nach einem Informatikstudium?* Jgg. P-219 in *LNI*. GI, S. 147–156, 2013.

- [Ha09] Hattie, John Allan Clinton: *Visible Learning. A synthesis of over 800 meta-analyses relating to achievement.* Routledge, London and New York, 2009.
- [Hu03] Humbert, Ludger: *Zur wissenschaftlichen Fundierung der Schulinformatik.* Dissertation, Witten, 3 2003. zugl. Dissertation an der Universität Siegen.
- [Ka12] Kaub, Kathrin; Kathrin, Julia; Biermann, Antje; Friedrich, Anja; Bedersdorfer, Hans-Werner; Spinath, Frank M.; Brünken, Roland: *Berufliche Interessensorientierungen und kognitive Leistungsprofile von Lehramtsstudierenden mit unterschiedlichen Fachkombinationen.* *Zeitschrift für pädagogische Psychologie*, 26(4):233–249, 2012.
- [Kl07] Klika, Dorle: *Fächerwahl im Lehramtsstudium – Zementierung der Geschlechtersegregation?* In (Schellack, Antje; Große, Stefanie, Hrsg.): *Bildungswege*, S. 123–133. Waxmann, Münster u. a., 2007.
- [Kl13] Kleinn, Karin; Götsch, Monika; Heine, Yvonne; Schinzel, Britta: *Das DFG-Projekt »Weltbilder der Informatik«.* *Informatik-Spektrum*, 36(3):251–256, 2013.
- [Kl15] Klemm, Klaus: *Lehrerinnen und Lehrer der MINT-Fächer: Zur Bedarfs- und Angebotsentwicklung in den allgemein bildenden Schulen der Sekundarstufen I und II am Beispiel Nordrhein-Westfalens*, 2015.
- [Kn11] Knobelsdorf, Maria: *Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung.* Dissertation, Freie Universität Berlin, 2011.
- [MS15] MSW-NW: *Das Schulwesen in Nordrhein-Westfalen aus quantitativer Sicht 2014/15. Statistische Übersicht 388.* Düsseldorf, 5 2015. MSW-NW – Ministerium für Schule und Weiterbildung Nordrhein-Westfalen.
- [Mü17] Müller, Dorothee: *Der Berufswahlprozess von Informatiklehrkräften.* *Commentarii informaticae didacticae (CID) 11*, Universitätsverlag Potsdam, Potsdam, 2017. Überarbeitete Fassung der Dissertation, Universität Wuppertal, 2016.
- [Pa13] Pant, Hans Anand; Stanat, Petra; Schroeders, Ulrich; Roppelt, Alexander; Siegle, Thilo; Pöhlmann, Claudia: *IQB-Ländervergleich 2012. Mathematische und naturwissenschaftliche Kompetenzen am Ende der Sekundarstufe I.* Waxmann, Münster, New York, 2013. IQB – Institut zur Qualitätsentwicklung im Bildungswesen.
- [PM16] Pieper, Monika; Marsching, Michele: *Schulministerin Löhrmann ermutigt Schülerinnen und Schüler zur Wahl des Fachs Informatik, doch wer soll sie unterrichten? Kleine Anfrage 4731 vom 2. Mai 2016. Antwort der Ministerin für Schule und Weiterbildung namens der Landesregierung.* Drucksache 16/11876, Landesregierung des Landes Nordrhein-Westfalen, Düsseldorf, 2016. Datum des Originals: 30.05.2016/Ausgegeben: 02.06.2016.
- [Ro14a] Rothland, Martin: *Warum entscheiden sich Studierende für den Lehrerberuf?* In: (Terhart et al.) [TBR14], S. 349–385.
- [Ro14b] Rothland, Martin: *Wer entscheidet sich für den Lehrerberuf?* In: (Terhart et al.) [TBR14], S. 319–348.
- [St14] Statistisches Bundesamt, Hrsg. *Bildung und Kultur – Prüfungen an Hochschulen. Prüfungsjahr 2013. Fachserie 11 Reihe 4.2.* Wiesbaden, 2014.
- [TBR14] Terhart, Ewald; Bennewitz, Hedda; Rothland, Martin, Hrsg. *Handbuch der Forschung zum Lehrerberuf*, Münster, New York, München, Berlin, 2014. Waxmann.
- [Ul04] Ulich, Klaus: *»Ich will Lehrer, -in werden« eine Untersuchung zu den Berufsmotiven von Studierenden.* Weinheim, 2004.

Mit dem Glauben Berge versetzen ... - Die Selbstwirksamkeitserwartung von Informatiklehrkräften

Claudia Hildebrandt¹

Abstract: Den Unterrichtsstoff sachlich richtig und interessant vermitteln, motivieren, optimal auf Prüfungen vorbereiten, Durchsetzungsvermögen zeigen, die Klasse im Griff haben, Konflikte offensiv angehen und bewältigen und vieles mehr wird von den Lehrerinnen und Lehrern heutzutage erwartet. Ob sie allerdings derartig hohen fachlichen und pädagogischen Anforderungen gerecht werden, ist von vielen Faktoren abhängig. Ein bedeutender Faktor ist die individuelle Überzeugung von den eigenen Fähigkeiten, die sogenannte Selbstwirksamkeitserwartung. Sie beeinflusst Wahrnehmungen, die Motivation, die Leistungen sowie das Handeln. Somit geht es im Folgenden um die Frage, welche Selbstwirksamkeitserwartung Informatiklehrkräfte bezogen auf den Informatikunterricht im Bereich der Sekundarstufe I vor und nach einer Fortbildungsmaßnahme haben. Es wird dabei nicht nur die allgemeine informatikspezifische Lehrerselbstwirksamkeitserwartung mit Hilfe von Fragebogenergebnissen von 41 Lehrkräften untersucht, sondern ebenfalls die informatikspezifische Lehrerselbstwirksamkeitserwartung für spezielle Themenbereiche wie algorithmisches Problemlösen, Daten und ihre Spuren und automatisierte Prozesse. Darüber hinaus ist von Interesse, ob die unterschiedlichen Selbstwirksamkeitserwartungen bezüglich dieser speziellen Themenbereiche mit der empfundenen Qualität einer Fortbildung in Zusammenhang stehen. Die Untersuchungsergebnisse bestätigen, dass eine Steigerung der informatikspezifischen Lehrerselbstwirksamkeitserwartungen durch bestimmte Arten von Fortbildungen möglich ist. Weiterhin weisen sie auf einen Zusammenhang zwischen der Qualität der Fortbildungen und der Lehrerselbstwirksamkeitserwartungen hin. Eine Entfaltung des Potenzials von Lehrkräften durch eine Steigerung der individuellen Lehrerselbstwirksamkeit mithilfe von Fortbildungen ist somit möglich, wovon letztendlich die Schülerinnen und Schüler profitieren.

Keywords: Selbstwirksamkeitserwartung, Lehrerselbstwirksamkeitserwartung, Lehrerfortbildung, Einfluss von Fortbildungen, empirische Untersuchung

1 Einführung und Motivation

Aufgrund der Tatsache, dass es, insbesondere in Niedersachsen, wenig Informatiklehrkräfte gibt, die durch das Studium des Faches Informatik mit dem Berufsziel Lehramt ihre Qualifikation zum Unterrichten dieses Faches erworben haben, ist es unabdingbar, durch geeignete Fortbildungsmaßnahmen dafür zu sorgen, dass ein zeitgemäßer curriculumskonformer Informatikunterricht ermöglicht wird. Dabei gilt es allerdings nicht außer Acht zu lassen, dass die Verwirklichung von gutem Unterricht im Sinne der Schülerinnen und Schüler von vielen Faktoren wie der individuellen Überzeugung über eigene Fähigkeiten, die sogenannte Selbstwirksamkeitserwartung [Ba97], abhängt. Im folgenden Kapitel wird somit zunächst das Konzept der Selbstwirksamkeit nach Albert Bandura vorgestellt und veröffentlichte Forschungsergebnisse zu diesem Thema dargestellt. Schlussfolgerungen

¹ Carl von Ossietzky Universität Oldenburg, Fakultät II - Didaktik der Informatik, 26111 Oldenburg, claudia.hildebrandt@uni-oldenburg.de

aus der Selbstwirksamkeitsforschung folgen für die selbst durchgeführten Lehrerfortbildungen in Kapitel 3. Die Forschungsfragen und notwendigen Definitionen werden in Kapitel 4 präsentiert und die Messinstrumente zur Erhebung der beschriebenen Konstrukte mithilfe der quantitativen Datenerhebung in Kapitel 5 erläutert. Kapitel 6 beschäftigt sich mit den ersten Ergebnissen. Abschließend werden die Resultate zu den Forschungsfragen diskutiert (siehe Kapitel 6) und ein Ausblick wird gegeben.

2 Stand der Forschung

Die Grundlage der folgenden Ausführungen bildet das von Albert Bandura [Ba77] entwickelte Konzept der Selbstwirksamkeit. Das ursprünglich im Rahmen der klinischen Psychologie entstandene Konzept hat in der empirischen Bildungsforschung zunehmend an Bedeutung gewonnen. Selbstwirksamkeitserwartungen sind optimistische Überzeugungen von der eigenen Fähigkeit, schwierige Anforderungssituationen mit den zur Verfügung stehenden Ressourcen erfolgreich bewältigen zu können [Ba77]. Bandura definiert die Selbstwirksamkeitserwartung als „beliefs in one’s capabilities to organize and execute the courses of action required to manage prospective situations“ [Ba95, S. 2]. Sie wirkt sich auf das Denken, Fühlen und unsere Motivation aus [Ba95, S. 3] und beeinflusst als ein Schlüsselfaktor unser Handeln. „Hence different people with similar skills, or the same person under different circumstances, may perform poorly, adequately, or extraordinarily, depending on fluctuations in their beliefs of personal efficacy“ [Ba77, S. 37].

Die Ergebnisse aus der Selbstwirksamkeitsforschung der letzten etwa 40 Jahre werden im Artikel *Teacher Self-Efficacy and Its Effects on Classroom Processes, Student Academic Adjustment, and Teacher Well-Being: A Synthesis of 40 Years of Research* von Marjolein Zee and Helma M. Y. Koomen [ZK16] zusammengefasst. Es zeigt sich, dass Lehrkräfte mit einer hohen berufsspezifischen Selbstwirksamkeit positiv mit der Qualität der Prozesse im Unterricht in Beziehung stehen. Sie sind eher in der Lage, Unterstützung beim Lernen zu geben, zeigen eine gute Unterrichtsorganisation und eine emotionale Unterstützung der Schülerinnen und Schüler. Diese Faktoren stehen wiederum mit den schulischen Leistungen und der Motivation der Lernenden in Beziehung (ebenfalls zu finden in [Ba97, S. 240 - 243], [HPK13], [GD84]). Aber nicht nur die Schülerinnen und Schüler profitieren von einer hohen berufsspezifischen Selbstwirksamkeit von Lehrkräften. Sie wirkt sich ebenfalls positiv auf die Gesundheit der Lehrerinnen und Lehrer und als Präventionsfaktor gegen zum Beispiel Burnout aus [SS00]. Dieses war eines der Ergebnisse des deutschen Modellversuchs *Selbstwirksame Schulen* von 1995 - 1998, an dem zehn Schulen mit ihrer Sekundarstufe teilnahmen. Ein gemeinsames Ziel war es, das Niveau der Selbstwirksamkeit bei Lehrkräften und Schülerinnen und Schülern zu erhöhen [SJ99].

Jetzt stellt sich die Frage, wie man die Selbstwirksamkeitserwartung entwickeln und erhalten kann. Bandura [Ba97, S. 79 - 115] beschreibt in diesem Zusammenhang vier Quellen:

- **Enactive master experience:** Eigene Erfolgserlebnisse stellen den einflussreichsten Faktor dar. Ein robuster Glaube an die individuelle Selbstwirksamkeit kann nur

durch die eigene erfolgreiche Bewältigung (schwieriger) Situationen erlangt werden. Misserfolge führen zu einer Reduzierung der Selbstwirksamkeitserwartung.

- **Vicarious experiences:** Andere Menschen mit Fähigkeiten, die den eigenen als ähnlich eingestuft werden, können zu Vorbildern werden. Meistern diese Vorbilder eine Aufgabe, traut man sich selbst auch eher das Lösen derartiger Aufgaben zu. Andererseits demotivieren Misserfolge.
- **Verbal persuasion:** Menschen strengen sich eher an und versuchen, die Herausforderungen zu meistern, wenn ihnen gut zugeredet und von anderen zugetraut wird, eine bestimmte Situation zu bewältigen.
- **Physiological and affective states:** Die Situations- und Selbstwirksamkeitsbewertung erfolgt unter anderem über die eigenen physiologischen Reaktionen wie Herzklopfen oder Schweißausbrüche in vor allem stressigen Situationen. Menschen mit hoher Selbstwirksamkeitserwartung interpretieren diese physiologischen Reaktionen als normal und unabhängig von den eigenen Fähigkeiten, wohingegen eine Interpretation als Schwäche Selbstzweifel aufkommen lassen könnte.

Die Anzahl der Forschungen zu Faktoren, die im schulischen Alltag die Selbstwirksamkeit von Lehrenden beeinflussen, ist bisher gering [Ge13, S. 9f]. Die Untersuchungen von Gebauer im Rahmen des *Modellvorhabens Selbstständige Schulen NRW* (von 2003 - 2008) ergaben, dass der affektive und emotionale Zustand den stärksten und stabilsten Zusammenhang zur Selbstwirksamkeitsüberzeugung von Lehrenden zeigt [Ge13, S. 139]. Darüber hinaus fanden Holzberger, Philipp und Kunter in ihrer Längsschnittstudie unter anderem heraus, dass die Selbstwirksamkeit von Lehrerinnen und Lehrern zu den Erfolgserlebnissen im Klassenraum in Beziehung steht - unabhängig von der Anzahl der Jahre an Unterrichtserfahrung [HPK13]. Ziel ist es nun, im Rahmen von Fortbildungen das vorhandene Potenzial der Lehrkräfte zu steigern und voll auszunutzen. Mit möglichen Konsequenzen, die aus den erwähnten Ergebnissen für Fortbildungen abgeleitet werden können, beschäftigt sich das nächste Kapitel.

3 Die Fortbildungen

3.1 Das Konzept

Eine bedeutende Schlussfolgerung, die man aus den Untersuchungsergebnissen (siehe Kapitel 2) ziehen kann, ist, dass der effektivste Weg, die Selbstwirksamkeit zu entwickeln und zu steigern, mithilfe von positiven, eigenen Erfahrungen erfolgt. Das heißt für Lehrerfortbildungen, dass die Praxisanteile deutlich höher als die Theorieanteile liegen sollten. Aber dabei gilt es zu beachten, dass die Aufgaben so gewählt werden müssen, dass die nach einer Phase der Anstrengung auch erfolgreich bewältigt werden können. Beispielsweise bei Fortbildungen zum Thema automatisierte Prozesse mithilfe von Lego Mindstorms Robotern können bei komplexen Aufgaben sowohl Fehler bei der Konstruktion des Roboters als auch Fehler in der Programmierung auftreten. Somit ist es notwendig, zunächst Aufgaben

zu wählen, die von jeder Teilnehmerin und jedem Teilnehmer erfolgreich bewältigt werden können. Dazu gehören einfache Fahraufgaben, ein Ballwerfer, der unter dem Einsatz eines „Wurfarmes“ mit einem kleinen Ball Kegel aus Legoteilen umwerfen soll. Es ist eine motivierende Aufgabe, bei der man den Erfolg direkt selbst sieht. Anschließend können Aufgaben gestellt werden, die immer komplexer werden, z.B. ein Quadrat abfahren, und schließlich auch das Sensor-Aktor-Prinzip berücksichtigen wie Hindernissen ausweichen und einer schwarzen Linie folgen. Ein weiterer Aspekt, der Berücksichtigung findet, ist das Lernen vom Vorbild. Nicht nur die Fortbilder können Vorbilder sein, indem sie von Lernwiderständen im Unterricht berichten und Möglichkeiten zur Bewältigung aufzeigen, sondern auch die anderen Fortbildungsteilnehmerinnen und -teilnehmer. Die Kolleginnen und Kollegen werden in erster Linie als Vorbilder angesehen, da die Ähnlichkeiten und Gemeinsamkeiten am stärksten sind. Somit kann es zu einer Beeinflussung durch diese Vorbilder kommen. Es gilt, den Austausch unter den an der Fortbildung teilnehmenden Personen explizit zu fördern und zu fordern. Darüber hinaus ist es immer wieder wichtig, den Lehrkräften ihre Erfolge vor Augen zu führen, z.B. durch Lob und Würdigung der Ergebnisse oder Teilergebnisse und der damit verbundenen Anstrengung sowie sie zu ermutigen, weiter zu machen, da sie die Fähigkeit zum erfolgreichen Bewältigen der Aufgabe besitzen. Damit sollen Selbstzweifel überwunden werden oder gar nicht erst aufkommen. Um stressvolle Situationen zu vermeiden, ist es sinnvoll, den Teilnehmerinnen und Teilnehmern Aufgaben unterschiedlichen Schwierigkeitsgrades anzubieten. Für die Fortgeschrittenen gibt es die Aufgaben ohne Hilfsanweisungen sowie weiterführende Aufgaben, für die Anfänger einfachere Aufgaben mit mehr Hilfestellungen. Individuelle Lernumgebungen können geschaffen werden, um einer Abnahme der Selbstwirksamkeitserwartung entgegenzuwirken.

Zusammenfassend kann man festhalten, dass lernende Lehrkräfte durchaus mit herausfordernden und für den Unterricht bedeutsamen Aufgaben konfrontiert werden können, die aber auch mit einem gewissen Maß an Anstrengung bewältigbar sein müssen. Kontinuierliche Misserfolge müssen vermieden werden.

3.2 Inhalte und Umfang

Es wurde versucht, die beschriebenen Aspekte bei den drei hier untersuchten Lehrerfortbildungen anzuwenden. Bei den ersten beiden untersuchten Fortbildungen handelt es sich um eintägige, bei der dritten um eine dreitägige Fortbildung. Die erste Fortbildung, die im November 2016 mit 16 Personen stattfand, beinhaltete den Einsatz von Arduino-Mikrocontroller-Boards im Informatikunterricht. Den Fortbildungsteilnehmerinnen und -teilnehmern wurden Arduino-Mikrocontroller-Boards vorgestellt und sie sollten selbstständig eine im Unterricht einsetzbare Unterrichtsreihe als Lernende bearbeiten. Die zweite Fortbildung mit fünf Teilnehmerinnen und Teilnehmern im Januar 2017 war eine schulinterne Lehrerfortbildung für eine Oberschule. Hier ging es um die Einführung in den Themenbereich automatisierte Prozesse mit Lego Mindstorms Robotern in der Sekundarstufe I. Der Kurs führte in die Grundlagen der Robotik sowie die graphische Programmierumgebung von Lego ein. Die dritte Fortbildung mit 20 Personen war eine dreitägige Fortbildung, wobei jeder Tag seinen thematischen Schwerpunkt hatte. Am ersten Tag ging es um

das Lernfeld algorithmisches Problemlösen, was mithilfe von Scratch näher beleuchtet wurde. Am zweiten Tag rückte das Lernfeld Daten und ihre Spuren (Aufbau von Netzwerken mit Schwerpunkt Internet, Datenaustausch in Netzwerken, Kennenlernen von Verschlüsselungsverfahren) in den Fokus. Den Abschluss am dritten Tag bildete der Bereich automatisierte Prozesse (ähnlich durchgeführt wie die zweite Fortbildung im Januar). Alle Fortbildungen hatten gemeinsam, dass sie einen hohen praktischen Anteil zum Ausprobieren von digital zur Verfügung stehendem Unterrichtsmaterial hatten. Darüber hinaus gab es immer wieder kurze Theoriephasen und am Ende wurde der Bezug zum niedersächsischen Kerncurriculum Informatik für die Schuljahrgänge 5-10 (KC) hergestellt. Damit sollte den Lehrkräften veranschaulicht werden, wie viel sie eigentlich schon gelernt haben, um den Anforderungen, die das KC an sie stellt, problemlos gerecht zu werden. Darüber hinaus wird in den untersuchten Fortbildungen den Lehrkräften ein umfangreiches digitales Paket von Unterrichtsmaterialien zur Verfügung gestellt, damit sie die Möglichkeit haben, ihre Kompetenzen in Heimarbeit auszubauen und auch keine Scheu davor haben, das neu Gelernte trotz mangelnder Erfahrung im Unterricht einzusetzen.

4 Forschungsfragen und Definitionen

Unter Einbeziehung der Ergebnisse aus den Kapiteln 2 und 3 sowohl in Bezug auf die Definition der zu messenden Konstrukte als auch in Bezug auf die mangelnden Forschungsergebnisse zu den Einflussfaktoren auf die Lehrerselbstwirksamkeit, werden folgende Forschungsfragen gestellt:

1. Inwiefern verändert sich die individuelle informatikspezifische Lehrerselbstwirksamkeitserwartung im Allgemeinen bzw. jeweils bezogen auf Teile der Lernfelder algorithmisches Problemlösen, Daten und ihre Spuren sowie automatisierte Prozesse im Laufe der untersuchten Fortbildung?
2. Inwiefern hängen die unterschiedlichen informatikspezifischen Lehrerselbstwirksamkeitserwartungen nach der Fortbildung von der Qualität der Fortbildung ab?

Die Grundlage der folgenden Definitionen bildet das bereits existierende Konstrukt der individuellen Lehrerselbstwirksamkeitserwartung. Im Sinne Banduras versteht man unter der **individuellen Selbstwirksamkeit von Lehrkräften** die Überzeugung einer Lehrkraft, mit ihren Fähigkeiten und Ressourcen eine herausfordernde berufliche Anforderungssituation zu meistern. Zum Beispiel basiert die Fähigkeit, Lernumgebungen zu schaffen, die die kognitiven Kompetenzen der Lernenden entwickeln, unter anderem auch auf der Lehrerselbstwirksamkeit [Ba97, S. 240]. Die Entwicklung der Skala zur Lehrerselbstwirksamkeitserwartung erfolgte bereits von Schwarzer und Jerusalem [SJ99, S. 60]. Dieses Konstrukt wird weiter spezifiziert und konkret auf den Bildungsbeitrag des Faches Informatik bezogen [Ni14]. Es wird die **individuelle informatikspezifische Lehrerselbstwirksamkeitserwartungsskala** entwickelt. Mithilfe eines kompakten Messinstruments soll die informatikspezifische Selbstwirksamkeit der Lehrer erhoben werden. Beispielsweise wird Folgendes abgefragt: *Ich weiß, dass mein Informatikunterricht Schüler zum reflektierten und verantwortungsbewussten Umgang mit Informatiksystemen befähigt*. Es finden

die Aspekte Berücksichtigung, die das Fach Informatik zum Bildungsbeitrag der allgemeinbildenden Schulen beitragen kann. Die Schülerinnen und Schüler sollen ein gewisses Grundwissen erwerben und anwenden, um reflektierte und verantwortungsbewusste Entscheidungen treffen zu können.

Die darüber hinaus entwickelte **informatikspezifische Lehrerselbstwirksamkeitserwartungsskala** jeweils bezogen auf die Lernfelder algorithmisches Problemlösen, Daten und ihre Spuren sowie automatisierte Prozesse beinhaltet die Überzeugungen einer Lehrkraft, mit ihren Fähigkeiten und Ressourcen eine herausfordernde berufliche Anforderungssituation speziell bezogen auf einen abgegrenzten Themenbereich zu meistern. Dazu wurden die Kompetenzanforderungen der verschiedenen Bereiche im bereits erwähnten niedersächsischen Kerncurriculum [Ni14] untersucht und als Items formuliert. Dabei finden allerdings nicht alle Kompetenzen, die mit einem Lernfeld entwickelt werden sollen, Berücksichtigung, sondern lediglich die Kompetenzen, die mithilfe der Inhalte der jeweiligen Lehrerfortbildungen entwickelt und gefördert werden können.

Schließlich ist noch die von den Lehrkräften empfundene **Qualität der Fortbildung** von Bedeutung. Hier sollen sich die Quellen der Selbstwirksamkeitserwartung widerspiegeln (siehe Kapitel 2). Mithilfe dieses Konstrukts sollen in kompakter Form zum Beispiel Aussagen zur Angemessenheit der Fortbildungsziele, über das Theorie-Praxis-Verhältnis (enactive master experience), die produktive Zusammenarbeit mit Kolleginnen und Kollegen (in Teilen vicarious experiences), über das Engagement der Fortbilder (in Teilen verbal persuasion) und über die vorherrschende Atmosphäre (physiological and affective states) getätigt werden. Eine Übersicht über die Konstrukte, die dazugehörige Anzahl der Items, die Reliabilitäten der Skalen (Cronbachs Alpha) sowie Beispielitems sind in Tabelle 1 zu finden.

5 Forschungsmethode

An den Umfragen zu den in Kapitel 3 beschriebenen Fortbildungen nahmen insgesamt 41 Lehrkräfte (12 Frauen, 29 Männer) teil. Fragebögen wurden zur Datenerhebung vor (zum Zeitpunkt t_0) bzw. nach der jeweiligen Fortbildung bzw. des jeweiligen Fortbildungstages (zum Zeitpunkt t_1) erhoben. Die meisten der teilnehmenden Lehrkräfte gaben an, sich Informatik selbst beigebracht oder durch Fortbildungen Qualifikationen erlangt zu haben. Lediglich sechs Personen hatten Informatik als Erst- oder Zweitfach oder als Erweiterungsfach studiert. Die Lehrerinnen und Lehrer notierten, zwischen 26 und 63 Jahre alt zu sein. Drei Lehrkräfte waren am Gymnasium oder an einer Schule mit gymnasialem Angebot tätig, alle anderen Lehrkräfte arbeiteten an Kooperativen Gesamtschulen, Realschulen, Hauptschulen, aber vor allem an Oberschulen (28 Lehrkräfte).

Die Untersuchung der im letzten Kapitel vorgestellten Konstrukte erfolgte anhand von Items. Die Antwortskalen der Items waren sechsstufig, von „1 = trifft gar nicht zu“ bis „6 = trifft völlig zu“ (Likert-Skala). Das bedeutet, dass die Lehrkräfte jeweils nach dem Grad ihrer Zustimmung bzw. Ablehnung zu den Items befragt wurden. Antworten auf einer Likert-Skala sind ordinal- beziehungsweise rangskaliert. Da den Umfrageteilnehmerinnen und -teilnehmern die verschiedenen Antwortmöglichkeiten durch eine äquidistante

Skala visualisiert wurden, wird im Folgenden das Intervallskalenniveau verwendet. Die Ergebnisse werden dadurch nur unwesentlich „verfälscht“ ([HSE10, S. 67], [AI78]). Tabelle 1 zeigt die Konstrukte, Beispielitems und die Cronbachs Alpha Werte. Bei den Konstrukten, bei denen die Werte für Cronbachs Alpha über 0,7 liegen, ist davon auszugehen, dass die jeweilige Gruppe von Test-Items als Messung einer einzelnen Variablen (hier der einzelnen Konstrukte) angesehen werden können. Da bei der Lehrerumfrage, bei der die informatikspezifische Lehrerselbstwirksamkeitserwartung bezogen auf das Lernfeld algorithmisches Problemlösen erhoben wurde, lediglich 19 gültige Datensätze vorhanden sind, ist ein Wert für Cronbachs Alpha von 0,628 gerade noch so akzeptabel. Ähnlich verhält es sich mit dem Cronbachs Alpha Wert vom Konstrukt Qualität der Fortbildung bezogen auf das Lernfeld Daten und ihre Spuren. Dort sind 17 gültige Datensätze zu finden.

Konstrukt	n	Beispielitem	α
individuelle informatikspezifische Lehrerselbstwirksamkeitserwartung (LSWE)	8	Ich bin in der Lage, bei Schülern ein grundlegendes Verständnis von der Informatik zu entwickeln.	t_0 : 0,913 t_1 : 0,882
informatikspezifische LSWE (algorithmisches Problemlösen)	5	Ich weiß, dass ich es schaffe, den Lernenden die Kompetenz zu vermitteln, eindeutige Anweisungen als ausführbares Programm zu implementieren.	t_0 : 0,795 t_1 : 0,628
informatikspezifische LSWE (Daten und ihre Spuren)	7	Ich bin in der Lage, bei Schülern ein Verständnis für die dem Internet zugrundeliegenden technischen Strukturen und Kommunikationswege zu entwickeln.	t_0 : 0,878 t_1 : 0,839
informatikspezifische LSWE (automatisierte Prozesse)	5	Ich bin mir sicher, dass ich auf individuelle Probleme der Schüler im Bereich automatisierte Prozesse gut eingehen kann.	t_0 : 0,884 t_1 : 0,876
<i>Qualität der Fortbildung</i> algorithm. Problemlösen: Daten und ihre Spuren: automatisierte Prozesse:	6	Das Verhältnis von Theorie- und Praxisanteilen wurde angemessen gewählt.	t_1 : 0,760 t_1 : 0,689 t_1 : 0,956

Tab. 1: Die Merkmale (Anzahl der Items n, Cronbachs Alpha α) der erfassten Konstrukte im Überblick

6 Ergebnisse

6.1 Forschungsfrage 1

Zur Untersuchung der ersten Forschungsfrage *Inwiefern verändert sich die individuelle informatikspezifische Lehrerselbstwirksamkeitserwartung im Allgemeinen bzw. jeweils bezogen auf Teile der Lernfelder algorithmisches Problemlösen, Daten und ihre Spuren sowie automatisierte Prozesse im Laufe der untersuchten Fortbildung?* werden die Konstrukte individuelle informatikspezifische Lehrerselbstwirksamkeitserwartung sowie die informatikspezifische Lehrerselbstwirksamkeitserwartung jeweils bezogen auf das algorithmische

Problemlösen, Daten und ihre Spuren und automatisierte Prozesse zu den Erhebungszeitpunkten t_0 und t_1 untersucht. Die Skalenmittelwerte und Standardabweichungen bezüglich dieser Konstrukte sind der Tabelle 2 zu entnehmen. Man erkennt, dass die Skalenmittelwerte der jeweiligen Selbstwirksamkeiten vor den jeweiligen Fortbildungen ungefähr bei vier auf einer Skala von eins bis sechs liegen. Das heißt, es herrscht eine zwar nicht sehr hohe, aber eine gewisse positive Lehrerselbstwirksamkeit vor. Im Bereich des algorithmischen Problemlösens ist diese am geringsten ausgeprägt, wohingegen die allgemein gehaltene individuelle informatikspezifische Lehrerselbstwirksamkeitserwartung zu Beginn der Fortbildungen am größten ist. Das positive Resultat dieser Untersuchung ist die Tatsa-

Konstrukt	m_0	m_1	$m_1 - m_0$	Wilcoxon-Test (p-Wert)
individuelle informatikspezifische LSWE	4,21 ($\sigma \approx 0,87$)	4,94 ($\sigma \approx 0,55$)	+0,73	0,000
LSWE (algorithmisches Problemlösen)	3,98 ($\sigma \approx 1,03$)	4,49 ($\sigma \approx 0,75$)	+0,51	0,104
LSWE (Daten und ihre Spuren)	4,22 ($\sigma \approx 0,88$)	4,85 ($\sigma \approx 0,62$)	+0,63	0,006
LSWE (automatisierte Prozesse)	4,10 ($\sigma \approx 1,08$)	4,66 ($\sigma \approx 0,86$)	+0,56	0,002
Qualität der Fortbildung (algorithmisches Problemlösen)	-	5,31 ($\sigma \approx 0,44$)	-	-
Qualität der Fortbildung (Daten und ihre Spuren)	-	5,45 ($\sigma \approx 0,45$)	-	-
Qualität der Fortbildung (automatisierte Prozesse)	-	5,57 ($\sigma \approx 0,50$)	-	-

Tab. 2: Skalenmittelwerte m und Standardabweichungen σ der Konstrukte

che, dass die Selbstwirksamkeit in allen Bereichen während der Fortbildung zugenommen hat. Betrachtet man zusätzlich noch die Ergebnisse des nichtparametrischen zweiseitigen Wilcoxon-Tests, so kann man folgern, dass die Unterschiede der Skalenmittelwerte vor und nach der Lehrerfortbildung bis auf den Bereich des algorithmischen Problemlösens signifikant sind ($p < 0,05$). Im Bereich des algorithmischen Problemlösens ist ganz knapp keine statistische Signifikanz feststellbar. Weitere Untersuchungen werden zeigen, ob es eventuell doch eine statistische Signifikanz gibt.

6.2 Forschungsfrage 2

Inwiefern hängen die unterschiedlichen informatikspezifischen Lehrerselbstwirksamkeitserwartungen nach der Fortbildung von der Qualität der Fortbildung ab? stellt die zweite Forschungsfrage dar. Zunächst kann man Tabelle 2 entnehmen, dass die Qualität der Fortbildungen von Lehrkräften im Durchschnitt mit 5,44 auf einer Skala von eins bis sechs sehr hoch bewertet wurde. Darüber hinaus zeigen sich zum Erhebungszeitpunkt t_1 , also jeweils nach den Fortbildungen, statistisch starke bis sehr starke Zusammenhänge zwischen der informatikspezifischen Lehrerselbstwirksamkeitserwartung aller drei hier betrachteten

Lernfelder mit der jeweils empfundenen Qualität der Fortbildungen (siehe Tabelle 3). Den Kommentaren der Lehrkräfte kann entnommen werden, dass besonders positiv der sehr hohe Praxisanteil und das gut aufbereitete, schülergerechte, digitale Unterrichtsmaterial hervorgehoben wurden. Lediglich einer Lehrkraft war die Fortbildung nicht tiefgründig genug und nur sehr vereinzelt wurde mitgeteilt, dass Teile der Fortbildungen zu einer Überforderung geführt hätten, da das Vorgehen nicht kleinschrittig genug gewesen sein sollte. Welche Richtung dieser Zusammenhang hat, kann man aus dem Spearmans Roh Korrelationskoeffizienten nicht ablesen. Um das herauszufinden, wären weitere Untersuchungen notwendig.

	Qualität der Fortbildung
LSWE (algorithmisches Problemlösen) t_1	0,801**
LSWE (Daten und ihre Spuren) t_1	0,650**
LSWE (automatisierte Prozesse) t_1	0,681**

Tab. 3: Korrelationen (Spearmans Roh, ** $p < 0,01$)

7 Diskussion und Ausblick

Aufgrund der positiven Untersuchungsergebnisse lässt sich vermuten, dass die zeitliche Struktur der Fortbildungen, das Verhältnis der Theorie- und Praxisanteile, das vorbereitete Material sowie der Umgang mit den Kolleginnen und Kollegen und die Vermittlung der Inhalte der Mehrzahl der teilgenommenen Lehrkräften zugesagt hat. In dieser angenehmen Lernatmosphäre war es möglich, die informatikspezifische Lehrerselbstwirksamkeit deutlich zu steigern. Es ist zu vermuten, dass die hohen Praxisanteile, die mehrfach lobend in den Kommentarfelder des Fragebogens erwähnt wurden, einen großen Teil zu dieser positiven Entwicklung beigetragen haben.

Zusammenfassend lässt sich festhalten, dass sich die informatikspezifische Lehrerselbstwirksamkeit der Informatiklehrkräfte bis auf im Bereich des algorithmischen Problemlösens signifikant erhöht hat. Damit ist, wenn man die Ergebnisse erfolgreicher Untersuchungen berücksichtigt (siehe Kapitel 2), davon auszugehen, dass damit auch ein Stück weit die Qualität des Informatikunterrichts steigt. Lernumgebungen zu verschiedensten Lernfeldern der Informatik können geschaffen werden, die die kognitiven Kompetenzen der Schülerinnen und Schüler entwickeln, fördern und fordern. Mithilfe des den Lehrkräften zur Verfügung gestellten Unterrichtsmaterials und der entsprechenden positiven Selbstwirksamkeitserwartung ist dieses Ziel in der Praxis umsetzbar. Bei den beschriebenen Ergebnissen muss man allerdings beachten, dass sie bisher auf einer Stichprobe von 41 Personen beruhen. Weitere Datenerhebungen - auch in Kontrollgruppen - sind notwendig. Darüber hinaus ist nach einem gewissen Zeitraum, z.B. nach ein bis zwei Jahren, zu untersuchen, ob die Fortbildungen nachhaltig die Lehrerselbstwirksamkeit und die Qualität des Informatikunterrichts positiv beeinflusst haben.

Damit sich Fortbildungsmaßnahmen in einer Veränderung des Unterrichts niederschlagen, sollte in Zukunft vermehrt darüber nachgedacht werden, mehrtägige Fortbildungen anzubieten, damit die Lehrkräfte Vorstellungen von den verschiedensten Bereichen der Infor-

matik bekommen und selbst die Erfahrungen sammeln, wie sie diese Lernfelder im praktischen Informatikunterricht einbringen können. Darüber hinaus hat sich gezeigt, dass es von Vorteil ist, wenn mehrere Kolleginnen und Kollegen einer Schule an ein und derselben Fortbildung teilnehmen können, um die positiven Eindrücke zusammen in den Schulalltag zu integrieren - auch bei auftretenden Widerständen, die erst gemeinsam überwunden werden müssen. In diesem Zusammenhang können weitere Forschungen zu fachgruppenspezifischen Lehrerselbstwirksamkeitserwartungen angestellt werden. Denn, „wenn es einen Glauben gibt, der Berge versetzen kann, so ist es der Glaube an die eigene Kraft“ (Marie Freifrau von Ebner-Eschenbach, 1830 - 1916).

Literaturverzeichnis

- [Al78] Allerbeck, K. R.: Messniveau und Analyseverfahren - Das Problem strittiger Intervallskalen. *Zeitschrift für Soziologie*, (3):199–214, 1978.
- [Ba77] Bandura, A.: Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84(2):191–215, 1977.
- [Ba95] Bandura, A.: Exercise of personal and collective efficacy in changing societies. In (Bandura, A., Hrsg.): *Self-efficacy in Changing Societies*. Cambridge University Press, Cambridge, S. 1–45, 1995.
- [Ba97] Bandura, A.: *Self-efficacy: The exercise of control*. Freeman, New York, 1997.
- [GD84] Gibson, S.; Dembo, M. H.: Teacher Efficacy: A Construct Validation. *Journal of Educational Psychology*, 76(4):569–582, 1984.
- [Ge13] Gebauer, M. M.: *Determinanten der Selbstwirksamkeitsüberzeugung von Lehrenden. Schulischer Berufsalltag an Gymnasien und Hauptschulen*. Springer VS, Wiesbaden, 2013.
- [HPK13] Holzberger, D.; Philipp, A.; Kunter, M.: How Teachers' Self-Efficacy Is Related to Instructional Quality: A Longitudinal Analysis. *Journal of Educational Psychology*, 105(3):774–786, 2013.
- [HSE10] Hussy, W.; Schreier, M.; Echterhoff, G.: *Forschungsmethoden in Psychologie und Sozialwissenschaften*. Springer, Berlin, 2010.
- [Ni14] Niedersächsisches Kultusministerium (Hrsg.): *Kerncurriculum für die Schulformen des Sekundarbereichs I Schuljahrgänge 5 - 10 Informatik, Niedersachsen*. Unidruck, Hannover, 2014.
- [SJ99] Schwarzer, R.; Jerusalem, M.: *Skalen zur Erfassung von Lehrer- und Schülermerkmalen. Dokumentation der psychometrischen Verfahren im Rahmen der Wissenschaftlichen Begleitung des Modellversuchs Selbstwirksame Schulen*. Berlin, 1999.
- [SS00] Schmitz, G. S.; Schwarzer, R.: Selbstwirksamkeitserwartung von Lehrern: Längsschnitbefunde mit einem neuen Instrument. *Zeitschrift für Pädagogische Psychologie*, 14:12–25, 2000.
- [ZK16] Zee, M.; Koomen, H. M. Y.: Teacher Self-Efficacy and Its Effects on Classroom Processes, Student Academic Adjustment, and Teacher Well-Being: A Synthesis of 40 Years of Research. *Review of Educational Research*, 86(4):981–1015, 2016.

Ein gamebasierter Ansatz zum Programmierunterricht in der Lehrerinnen- und Lehrerbildung

Nico Steinbach, Eckart Zitzler

Abstract: Mit dem Lehrplan 21 hält in den deutsch- und mehrsprachigen Kantonen der Schweiz der Themenbereich Informatik sowie die Kompetenzorientierung Einzug in die Volksschule. Dadurch ergeben sich neue Herausforderungen für die Schule, den Unterricht sowie die Aus- und Weiterbildung von Lehrpersonen. Zudem gibt es für diesen Themenbereich derzeit keine offiziellen Lehrmittel. Die pädagogischen Hochschulen stehen vor der Aufgabe, die aktiven Lehrpersonen, neben anderen Lehrplan-21-Weiterbildungen, für den Informatikunterricht zu befähigen.

In diesem Zusammenhang gilt es ein didaktisches Konzept zu entwerfen, welches den folgenden Anforderungen gerecht wird: 1) Einen motivierenden und spielerischen Zugang zu Informatikkonzepten zu ermöglichen 2) Die Heterogenität der Teilnehmenden hinsichtlich des Vorwissens und der Zielstufe zu berücksichtigen, 3) Die Erwachsenenbildung und den Schulunterricht gemeinsam zu denken, 4) Den Bezug zu anderen Fächern aufzuzeigen.

Der vorliegende Beitrag beschreibt die Entwicklung und Durchführung von kompetenzorientierten Weiterbildungen zum Themenfeld Programmieren im Kanton Bern. Im Zentrum des Ansatzes steht ein Jump'n Run-Computerspiel, welches blockbasiert entwickelt und individuell ausgestaltet werden kann.

Keywords: Computerspiel, Lehrerinnen- und Lehrerbildung, Programmierunterricht, Lehrplan 21

1 Einleitung

Mit dem Projekt Lehrplan 21 wurde in der Schweiz im Zeitraum 2010-2015 ein gemeinsamer Lehrplan für die 21 deutsch- und mehrsprachigen Kantone entwickelt mit der Absicht, die Ziele der Volksschule zu harmonisieren. Eine wesentliche Neuerung ist, dass die Inhalte kompetenzorientiert formuliert und auf Zyklen verteilt sind (Zyklus 1 = Kindergarten bis zweite Klasse; Zyklus 2 = dritte bis sechste Klasse; Zyklus 3 = siebte bis neunte Klasse). Des Weiteren wurde erstmalig die Informatik in allen Zyklen verankert, wobei auch die Bezüge zu anderen Fächern thematisiert werden. Sie umfasst neben reinen Anwendungskompetenzen drei Kompetenzen zu den Themenfeldern Datenstrukturen, Algorithmen und Informatiksystemen, die jeweils in mehreren Kompetenzstufen ausdifferenziert sind. Je nach Kanton sind separate Zeitgefäße hierfür vorgesehen, die zudem vier Kompetenzen im Medienbereich umfassen. Im Kanton Bern handelt es sich um jeweils eine Wochenlektion im 5., 6., 7. und 9. Schuljahr, deren Umsetzung im Schuljahr 2018/19 beginnt.

Da derzeit keine entsprechenden Informatik-Lehrmittel zur Verfügung stehen und den meisten im Berufsfeld tätigen Lehrpersonen sowohl die fachlichen als auch die

fachdidaktischen Hintergründe fehlen, stehen die Schulen vor großen Herausforderungen. Für die Pädagogischen Hochschulen heisst das, eine große Anzahl von Lehrerinnen und Lehrern in kurzer Zeit – im Kanton Bern sind es gut 750 Personen – innerhalb von drei Jahren für den Informatikunterricht vorzubereiten, neben den anderen Weiterbildungen im Kontext des Lehrplan 21. Dabei ergeben sich diverse Fragen, die sich bei der Entwicklung von Weiterbildungsangeboten stellen: 1. Wie kann der Spanne zwischen keinem Vorwissen und bereits eigener Unterrichtserfahrung begegnet werden?; 2. Wie kann das zu erarbeitende Konzept inklusive Material aufbereitet werden, damit es sowohl für die Lehrerinnen- und Lehrerbildung als auch für den Unterricht genutzt werden kann?; 3. Wie kann der Bezug zu anderen Fächern aufgezeigt und implementiert werden?

An der Pädagogischen Hochschule Bern wurde ein einwöchiger Blockkurs für Volksschullehrpersonen entwickelt, in dem sowohl die Medien- als auch die Informatikkompetenzen behandelt werden. Dieser Beitrag fokussiert die Algorithmik und diskutiert, wie das Themenfeld der Programmierung innerhalb eines Tages in diesen Blockkurs eingebettet wurde. Im Folgenden wird die Entwicklung eines entsprechenden didaktischen Konzepts vorgestellt und die Umsetzung im Schulunterricht sowie in der Weiterbildung diskutiert.

1.1 Hintergrund

Im Abschlussbericht der Arbeitsgruppe ICT und Medien, die das Modul Medien und Informatik des Lehrplan 21 ausgearbeitet hat, wird für die Weiterbildung von amtierenden Lehrpersonen ein Umfang von mindestens 3 ECTS-Punkten (1 ECTS-Punkt entspricht 30 Stunden) für Primarlehrpersonen bzw. 5 ECTS-Punkten für Sekundarlehrpersonen empfohlen [MI15]. Aufgrund verschiedener praktischer Rahmenbedingungen steht den Lehrpersonen jedoch in der Regel wesentlich weniger Zeit zur Verfügung, um sich im Themenfeld aus- und weiterzubilden.

1.2 Rahmenbedingungen im Kanton Bern

Im Kanton Bern ist die Weiterbildung zum Modul Medien und Informatik nicht obligatorisch. Es liegt in der Verantwortung der Schulleitungen zu bestimmen, welche Lehrpersonen die spezifischen Lektionen in der 5., 6., 7. und 9. Klasse unterrichten werden. Weil zudem eine Vielzahl weiterer, obligatorischer Weiterbildungen im Kontext des Lehrplan 21 absolviert werden müssen, die mit ca. 12 Stunden pro Fach dotiert sind, sind die zeitlichen Ressourcen für das Themenfeld Medien und Informatik begrenzt. In Absprache mit der Erziehungsdirektion wurde ein 30-stündiges, einwöchiges Weiterbildungsformat entwickelt, wobei drei Tage den Medien-Kompetenzen und zwei Tage den Informatik-Kompetenzen gewidmet sind. In diesen Blockwochen mit knapp 50 Teilnehmenden sind sowohl Primarlehrpersonen (betrifft 5. und 6. Klasse) als auch Sekundarlehrpersonen (betrifft 7. und 9. Klasse) weiterzubilden. Aufgrund der

beschränkten personellen Ressourcen sind die Teilnehmendengruppen gemischt, zudem bringen die Teilnehmenden sehr unterschiedliche persönliche Erfahrungen zum Thema Medien und Informatik mit.

Aufgrund dieser Rahmenbedingungen ist die Zielsetzung des Blockkurses vor allem den Lehrpersonen die fachlichen Hintergründe zum Modul Medien und Informatik zu vermitteln, fachdidaktische Ideen aufzuzeigen und Motivation und Interesse für das Thema zu wecken, damit diese sich selbständig weiter mit dem Thema beschäftigen können. Insbesondere ergeben sich folgende Anforderungen an das Kurskonzept: 1) Es kann Teilnehmende unterschiedlichen Vorwissens abholen und ins Thema einführen beziehungsweise weiterführende Themen anbieten; 2) Es ist geeignet für die Lehrerinnen- und Lehrerbildung als auch den anschließenden Einsatz im Unterricht; 3) Es zeigt konkrete Verknüpfungen zu anderen Fachbereichen auf; 4) Es spricht die Kreativität an und motiviert durch Lebensweltbezug.

Die Entwicklung solcher Materialien und Konzepte obliegt normalerweise den kantonalen Lehrmittelverlagen und Lehrmittelaufgebern. Eine Weiterbildung kann sich dann auf die vorgeschriebenen Lehrmittel beziehen. Zum Zeitpunkt der ersten Durchführung des Blockkurses lag jedoch kein entsprechendes Lehrmittel vor. Aus diesem Grund wurde die Entwicklung eines eigenen Konzepts angestrebt. Im Folgenden fokussieren wir uns auf den Aspekt des Programmierunterrichts – ein Tag im Blockkurs –, der im Lehrplan unter der Kompetenz Algorithmen beschrieben wird: “Die Schülerinnen und Schüler können einfache Problemstellungen analysieren, mögliche Lösungsverfahren beschreiben und in Programmen umsetzen.” [LP16] Die Kompetenz unterteilt sich in neun Kompetenzstufen, die mehrheitlich im zweiten Zyklus liegen. Die Frage, mit der sich dieser Beitrag auseinandersetzt, lautet, wie eine eintägige Weiterbildung zum Thema Programmierung didaktisch konzipiert und umgesetzt werden kann, sodass die oben aufgeführten Anforderungen erfüllt werden können.

1.3 Einbettung

Es existiert eine Vielzahl von Ansätzen und entsprechender Erfahrung zum Programmierunterricht in der Schule. Neben dem Programmieren ohne Computer können vor allem die textbasierte und die visuelle Programmierung genannt werden. Die textbasierte Programmierung blickt auf eine langjährige Debatte nach der Wahl einer geeigneten Programmiersprache, nebst Programmierparadigma zurück. Das Schreiben von textbasierten Programmen kann Schülerinnen und Schülern Schwierigkeiten bereiten, obwohl die Motivation und Fähigkeit zum algorithmischen Problemlösen vorhanden ist [MMS11]. Eine mögliche Konsequenz ist, dass das Interesse an der Informatik sinkt [Hu00]. Als Argumente gegen die textbasierte Programmierung werden unter anderem aufgeführt: Die Kodierung des Programmcodes in Textform ist nicht intuitiv [MMS11] und Menschen können Bilder viel früher verstehen als Text [Sc01]. Für die Förderung algorithmischen Denkens bietet die visuelle Programmierung für Anfänger Vorteile. Sie erlaubt es, sich schneller auf den Inhalt zu fokussieren und

vermeidet Syntaxfehler [Mo11].

Mit Scratch – die offizielle erste Version wurde 2007 veröffentlicht – existiert eine solche visuelle Programmierlernumgebung, die sich direkt an Kinder und Jugendliche richtet, blockbasierte Entwicklung von Programmen ermöglicht und Spiele ins Zentrum stellt. Neben Weiterentwicklungen (Scratch 2.0, 2013) und Modifikationen (BYOB/Snap) gibt es auch weitere Plattformen wie code.org [CO13], die auf die Kombination aus Blöcken und Spielen setzen. Mit der Bewegung Hour-of-Code [HO13] und prominenter Unterstützung (u.a. Barack Obama, Mark Zuckerberg, Bill Gates) schaffte es die Programmierung in die US-Schulen.

Der Einsatz von Spielen, wie er auch bei code.org verfolgt wird, scheint vielversprechend. Allein bis Ende 2015 konnte code.org über 100 Millionen Schülerinnen und Schüler auf der ganzen Welt erreichen [US15]. Auch die Bitkom konnte aufzeigen, dass sich Gaming in allen Altersgruppen etabliert hat [BI15]. Spiele sind dankbare Vehikel für Lerninhalte [Pr01]. Allerdings sind die zugrundeliegenden didaktischen Ansätze nur begrenzt auf das oben skizzierte Weiterbildungsszenario übertragbar. Die Aufgaben bei code.org schränken die Lernenden ein, indem sie die zu nutzenden Blöcke und damit das Wissen zur Lösung einer Aufgabe vorgeben. Schon Romeike kritisierte, dass Problemstellungen (Start, Ziel, Hürde) in der Informatik eher als Aufgaben (vorgegebenes Wissen und definierte Lösung) formuliert werden. Die Kreativität die der Informatik immanent ist, kommt nicht zur Entfaltung [Ro08]. Zusätzlich erschwert ein solches Setting die Fächerverbindung, da es kaum Ansatzpunkte gibt, Inhalte anderer Fächer einzubinden.

2 Didaktischer Ansatz

In diesem Beitrag wird die Idee verfolgt, die Lernenden ein Computerspiel (weiter)entwickeln zu lassen und so die Kompetenzstufen im Bereich der Algorithmik abzudecken. Beim Computerspiel handelt es sich um ein einfaches Jump'n-Run-Computerspiel, in dem eine Spielfigur von links nach rechts durch Spielumgebungen geführt werden kann. Eine Grundfunktionalität soll zur Verfügung gestellt werden, die dann von den Lernenden schrittweise erweitert wird, z. B. indem die Spielfigur neue Fähigkeiten erhält oder die Spielumgebung ausgebaut wird.

Konzept. Das didaktische Konzept, das an der PHBern entwickelt wurde, basiert auf drei Eckpfeilern:

1. Eine blockbasierte Programmierumgebung, die einen möglichst einfachen Zugang zum Programmieren ermöglicht, den Austausch mit anderen Programmierenden fördert, frei verfügbar ist und zudem erlaubt, Multimedia-Inhalte niederschwellig einzubinden; Letzteres war uns wichtig, um die inhaltliche Anbindung an andere Fächer zu erleichtern.

2. Ein erweiterbares Programmgerüst, welches die Grundfunktionalität des Computerspiels implementiert, gleichzeitig aber möglichst schlank gehalten ist, damit die Lernenden das Spiel erweitern und abändern können. Der Nachteil eines solchen vorgegebenen Systems ist, dass die Lernenden sich überfordert und verloren fühlen können. Für uns stand jedoch aufgrund der Heterogenität der Lerngruppen im Vordergrund, dass früh sichtbare Erfolgserlebnisse und zudem sehr offene Aufgabenstellungen möglich sind.
3. Eine interaktive Wegleitung, anhand derer die Lernenden das Computerspiel individuell weiterentwickeln können. Sie besteht aus mehreren multimedialen Aufgabenblättern. Jedes Aufgabenblatt enthält eine Sequenz von Aufgaben, die durch Zusatzinformationen (Videos, Texte etc.) und Beispiele angereichert sind. Die insgesamt neun Aufgabenblätter decken alle Kompetenzstufen im Bereich Algorithmik ab.

Wir haben uns bewusst dafür entschieden, die Lernenden mit einer vollständigen Programmierumgebung zu konfrontieren und die Möglichkeiten der Programmierung in Abhängigkeit der Aufgabenstellung explizit nicht einzuschränken. Des Weiteren folgt die Konzeption der Aufgabenblätter der Idee der Lernspirale [Ki13]: Anstatt fundamentale Programmierkonzepte einzeln zu behandeln, werden Konstrukte wie Anweisungen, bedingte Anweisungen, Schleifen und Variablen in allen Aufgabenblättern benutzt. Die Lernenden begegnen den elementaren Ideen also immer wieder, allerdings nimmt die Komplexität der Aufgaben zu. Damit einher geht auch das Konzept, dass die Aufgabenblätter nicht linear, sondern eher baumartig angeordnet sind. Einzig die zwei ersten Aufgabenblätter sind hintereinander zu bearbeiten, danach können die Lernenden nach ihren Fähigkeiten und Interessen auswählen.

Implementierung. Als Programmierumgebung wurde Scratch [SC17] gewählt. Scratch erfüllt die oben beschriebenen Anforderungen, hat allerdings (derzeit) den Nachteil, dass die Umgebung auf Geräten ohne Flash nicht einsetzbar ist; diese Einschränkung soll gemäß Ankündigung der Scratch-Entwickler behoben werden [SC16]. Das Programmgerüst, um die Grundfunktionalität des Spiels zu realisieren, umfasst fünf Funktionen und insgesamt knapp vierzig Einzelanweisungen; es definiert die Spielfigur sowie deren Bewegung über die Pfeiltasten und wird in Abbildung 1 skizziert.

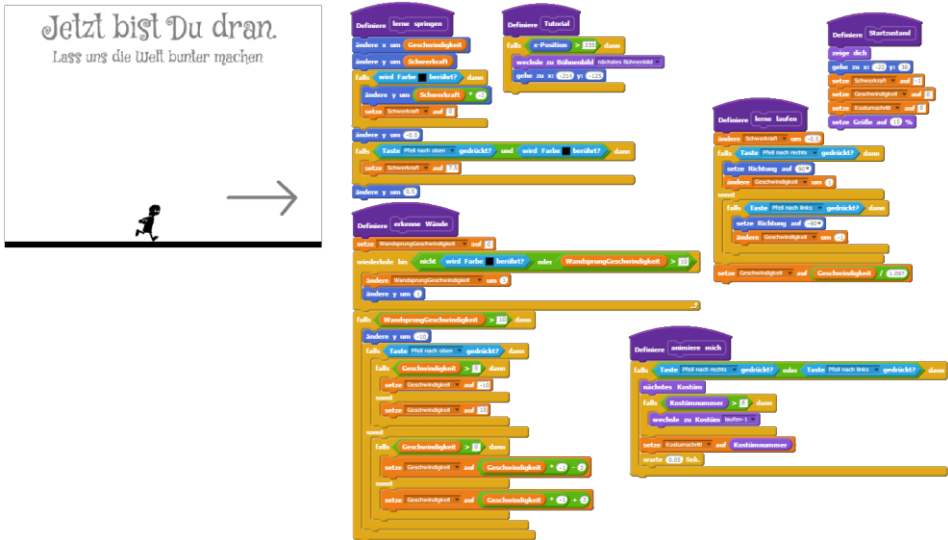


Abbildung 1: Das Spielfeld (links) und das Programmgrundgerüst in Scratch (rechts)

Die Arbeitsumgebung umfasst drei Einstiegshilfen in Scratch, neun Arbeitsblätter sowie einen Remixbaum (alle Weiterentwicklungen unseres Ursprungprojektes, die öffentlich sind), die in Form einer Baumstruktur auf einer Webseite verlinkt sind. Die vollständigen Materialien finden sich unter <https://milehrplan.ch/scratch>.

Die Arbeitsblätter werden on-the-fly aus einem JSON-Format generiert, welches in einer relationalen Datenbank liegt. Das ermöglicht sowohl eine Ansicht für Schülerinnen und Schüler ohne Lehrerkommentare, als auch eine weitere Ansicht für Lehrpersonen, zu erzeugen. Die Arbeitsblätter setzen sich wie folgt zusammen: Zwei fundamentale Arbeitsblätter (*“Der Beginn”*, *“Lass uns die Welt bunter machen”*) mit denen begonnen wird. Sie greifen niederschwellig einfache Anweisungen, bedingte Anweisungen sowie Schleifen auf und dienen dazu, der Spielfigur erste Interaktionen mit ihrer Umgebung zu ermöglichen, indem sie auf Farben reagiert. Aus den restlichen sieben Arbeitsblättern kann frei gewählt werden. Die drei fachübergreifenden Arbeitsblätter (*“Level Design”*, *“Story Design”*, *“Sound Design”*) dienen zur Erzeugung von Spielwelten (Bildnerisches Gestalten), Geschichten und Anleitungen (Deutsch) sowie Hintergrundmusik und Soundeffekten (Musik). Das erzeugte Material kann über bedingte Anweisungen in das Computerspiel integriert werden. Vier weitere Arbeitsblätter routinieren die Verwendung der eingeführten Konzepte und führen neue ein. Es handelt sich um Variablen (*“Spielement - Sammeln”*), Nachrichten zwischen Objekten (*“Spielement - Hindernis”*), Ereignisse (*“Spielement - Aktion”*) sowie die Implementierung weiterer Objekte wie zum Beispiel Gegner mit zufallsbasierten Bewegungspfaden (*“Spielement - Gegner”*).

Abbildung 2 stellt zwei Programmblöcke gegenüber – von einer früheren und einer

späteren Aufgabe – und vermittelt einen Eindruck vom Anforderungsniveau und der Komplexitätszunahme. Im linken Programmblock wird der Figur beigebracht, auf verschiedene Farben zu reagieren und entsprechend einfache Anweisungen auszuführen. Die Schleife simuliert den Gameloop, der nötig ist, um Interaktionen bis zum Programmende am Leben zu erhalten. Im rechten Programmblock wird entschieden, was mit einem zusätzlichen Objekt (Gegner) passiert, wenn die Variable *“LebenVirus”* auf unter 1 fällt. Dieser Programmblock ist nur ein Teil einer möglichen Lösung und interagiert mit weiteren Blöcken, welche die Variable ändern dürfen und Nachrichten schicken (nicht im Bild enthalten).

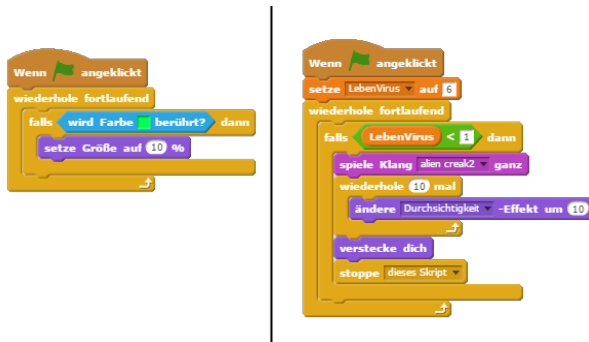


Abbildung 2: Gegenüberstellung zweier Programmblöcke unterschiedlichen Niveaus

3 Erfahrungen

Das vorgestellte Konzept wurde einerseits in Schulklassen getestet und andererseits in der Weiterbildung eingesetzt.

Der Einsatz im Unterricht fand in vier Klassen des siebten Schuljahres einer Sekundarschule statt mit je 16-18 Schülerinnen und Schülern. Die Erprobung dauerte über sechs Wochen mit je einer Lektion pro Woche. Stellvertretend für die Klassenlehrperson übernahm ein Mitarbeiter der PHBern die Rolle der Lehrperson, zwei weitere Mitarbeiter waren als Beobachter im Klassenraum. Das Ziel der sechs Wochen war die fachübergreifenden Themen zu erreichen, da die gesamte Einheit für bis zu 16 Lektionen ausgelegt ist. Im Vorfeld zu jeder Lektion entschied die Lehrperson, welche Arbeitsblätter zur Verfügung stehen. Anschließend haben sich die Schülerinnen und Schüler, meist zu zweit, selbstständig an die Inhalte gewagt.

In der allgemeinen Beobachtung und persönlichen Gesprächen fiel auf, dass das Computerspiel als Thema von beiden Geschlechtern sehr gut angenommen wurde. Die Arbeitsblätter wurden in unterschiedlichem Tempo im Pair Programming bearbeitet. Vor allem schnelle Teams halfen den anderen mit Hinweisen. Viele Teams haben intuitiv begonnen, aus den Problemstellungen eigene Projekte zu definieren. Alle Teams hatten

völlig individuelle Ergebnisse, durchliefen aber dieselben Konzepte. Auch über die Unterrichtszeit hinaus gab es zusätzliches Engagement in den Pausen und in der Freizeit. Die Lehrperson hat bei Fragen vor allem zwischen den Teams vermittelt und Konzepte im Plenum zusammengefasst. Am Ende der sechs Wochen wurden die entstandenen Projekte auf Initiative der Schülerinnen und Schüler abschließend vorgestellt.

Einschränkend war der 45-Minuten-Rahmen der pro Woche zur Verfügung stand. In der Regel benötigten die Schülerinnen und Schüler das erste Unterrichtsdrittel, um am Stand der letzten Woche anzuknüpfen. Ebenfalls kann es für die Lehrperson mitunter schwierig sein, an die unterschiedlichen Projekte und Lernstände anzuknüpfen nach einer Woche Pause zwischen den Lektionen.

In der Lehrerinnen- und Lehrerbildung haben bisher 95 Lehrpersonen die Blockwoche absolviert. An der ersten Blockwoche nahmen 48 Lehrpersonen teil, an der zweiten 47. Die Teilnehmenden wurden jeweils auf zwei Räume verteilt sowie zwei Dozenten zugeteilt. Die Kurstage erstreckten sich von 8:30 Uhr bis 17:00 Uhr. Zu Beginn des Algorithmentages wurden die Teilnehmenden in den Räumen in Vierer-Gruppen aufgeteilt. Anschließend wurde die Relevanz der Informatik für die Gesellschaft und Schule thematisiert und im Plenum die Vorerfahrung der Teilnehmenden diskutiert, bevor die selbstständige Arbeit begann.

Positiv hat sich gezeigt, dass die Thematik Computerspiele entgegen genereller Vermutungen auch bei Lehrpersonen Anklang findet. Zum einen, weil ihre Schülerinnen und Schüler oft über Games erzählen und sie motiviert sind, ihnen das Material vorzustellen. Zum anderen, weil sie Scratch und das Material als kreativ wahrnehmen und aktiv gestalten - im Gegensatz zu ihrer Vorstellung vom reinen Spielen kommerzieller Computerspiele. Selbst Lehrpersonen, die noch nie programmiert haben und/oder noch nie gespielt haben, nahmen den Ansatz dankend auf. Wie auch bei den Schülerinnen und Schülern gab es bei den Teilnehmenden unterschiedliche Arbeitstempi. Sie arbeiteten ebenfalls im Pair Programming und fingen an, eigene Projekte und kreative Umsetzungen aus den Arbeitsblättern zu generieren. Alle Teilnehmenden erreichten gegen Mittag die fächerübergreifenden Themen und nutzten den Nachmittag, neue Konzepte in den anderen Arbeitsblättern zu streifen.

Kritisch muss gesagt werden, dass an einem Kurstag keine echte Algorithmik erlernt werden kann und die Teilnehmenden die meisten Konzepte nur ausprobieren können. Die meisten reflektieren das für sich und geben an, sich weiterhin nicht sicher zu fühlen, das Thema im Unterricht aufzugreifen. Für sie bildet das Konzept nebst Material einen soliden Ausgangspunkt.

In einer abschließenden Befragung zu der gesamten Blockwoche wurde der Kenntnisstand in der Informatik vor und nach der Woche erfragt, siehe Abbildung 3.

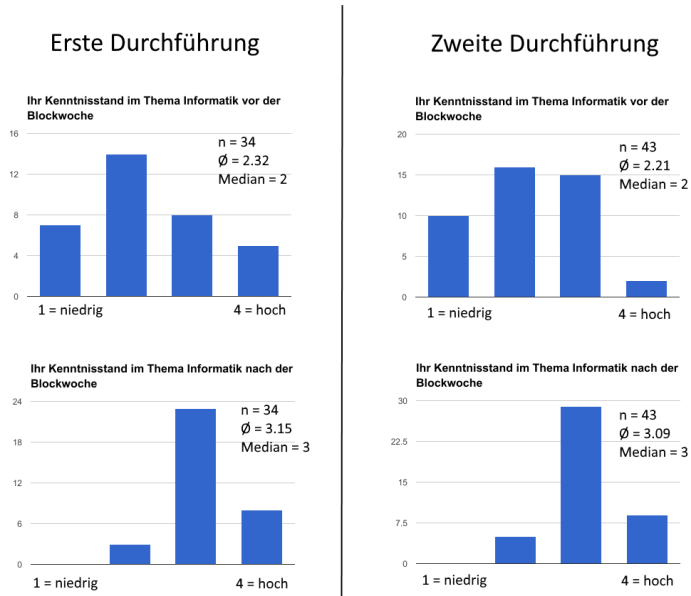


Abbildung 3: Kenntnisstand zur Informatik vor und nach den Blockwochen (Selbsteinschätzung)

Die in Abbildung 3 gezeigten Resultate beziehen sich auf die beiden Informatiktage der Woche zu den Themen Algorithmen, Datenstrukturen und Informatiksysteme. Es zeigte sich nach der Blockwoche, dass sowohl ein deutlicher Lernzuwachs zu verzeichnen ist, als auch dass sich keine Lehrperson nach der Woche selbst mit einem niedrigen Kenntnisstand einstuft.

4 Fazit

Das vorliegende didaktische Konzept ist geeignet um mit heterogenen Gruppen bestehend aus Lehrpersonen in der Lehrerinnen- und Lehrerbildung oder Schülerinnen und Schülern im Unterricht zu arbeiten. Ebenso vermag die Thematik eines Computerspiels beide Gruppen gleichermaßen anzusprechen und bietet in Verbindung mit einer visuellen Programmierlernumgebung wie Scratch eine niedrige Einstiegshürde für Programmieranfänger. Zudem eignet sich der Kontext rund um Spiele besonders gut, um kompetenzorientierte Problemstellungen zu formulieren, verschiedene Lösungswege zuzulassen und Raum für kreative Weiterentwicklungen zu ermöglichen. Das Konzept stellt hohe Anforderungen an Dozierende und Lehrpersonen und setzt voraus, dass sich Lernende gegenseitig unterstützen können.

Offen bleibt die Frage, wie nachhaltig eine eintägige Einführung in die Algorithmik sein

kann, schließlich lässt sich an einem Tag nur ein Einblick geben. Die Lehrpersonen gaben an, dass sie sich nach dem Tag nicht sicher fühlen, das Thema zu unterrichten, mit dem Material und der exemplarischen Durchführung jedoch eine solide Basis zum Anfangen gegeben ist. Die selbstständige Auseinandersetzung sowie weitergehende Weiterbildungsangebote sind eine Voraussetzung, damit das Thema effektiv in den Schulen unterrichtet werden kann. Erst dann kann sich ein didaktischer Diskurs bilden.

Literaturverzeichnis

- [BI15] Gaming hat sich in allen Altersgruppen etabliert, <https://www.bitkom.org/Presse/Presseinformation/Gaming-hat-sich-in-allen-Altersgruppen-etabliert.html>, Stand:14.02.2017
- [CO13] Code.org - Jeder Schüler sollte die Möglichkeit haben, Informatik zu lernen, <http://code.org>, Stand: 14.02.2017
- [Hu00] Bericht zur Lehrerausbildung Informatik, <http://koenigstein.inf.tu-dresden.de/00/humbert2.html>, Stand: 14.02.2017
- [HO13] Hour of Code, <https://hourofcode.com/de>, Stand: 14.02.2017
- [Ki13] Klippert, Heinz, Leitfaden zum Arbeiten mit Lernspiralen, 2013
- [LP16] Kompetenz Algorithmen im Modullehrplan Medien und Informatik, <http://v-ef.lehrplan.ch/index.php?code=a10|02|02>, Stand: 14.02.2017
- [MI15] Schlussbericht der Arbeitsgruppe zu Medien und Informatik im Lehrplan 21, https://www.lehrplan.ch/sites/default/files/Schlussbericht_MI_2015-02-23%20mit%20Anhang_0.pdf, Stand: 14.02.2017
- [MMS11] Modrow Eckart, Jens Mönig, Kerstin Strecker, LOG IN Heft Nr. 168, 2011
- [Mo11] Modrow, Eckart, Visuelle Programmierung: - oder: Was lernt man aus Syntaxfehlern?, Lecture Notes in Informatics, Proceedings, Volume P-249, 2011
- [Pr01] Prensky, Marc, Digital Game- Based Learning, McGraw-Hill, 2001
- [Ro08] Romeike, Ralf, Kreativität im Informatikunterricht, Mathematisch-Naturwissenschaftliche Fakultät Potsdam, 2008
- [SC17] Scratch - imagine, program, share - <https://scratch.mit.edu>, Stand: 14.02.2017
- [Sc11] Schiffer Stefan, Visuelle Programmierung, Grundlagen und Einsatzmöglichkeiten, 2001
- [SC16] Scratch-Blöcke (Google Blockly und Scratch - Open Source Projekt), <https://scratch.mit.edu/developers>, Stand: 14.02.2017
- [US15] Hour of Code to feature Star Wars: The Force Awakens. In: USA TODAY ,9. November 2015

Empirische Ermittlung der Schlüsselkonzepte des Fachgebiets Datenmanagement

Andreas Grillenberger,¹ Ralf Romeike²

Abstract: Die Aufbereitung neuer Themen für den Unterricht stellt eine wichtige Aufgabe der Informatikdidaktik dar. Diese schließt die Suche nach den zentralen Aspekten mit ein, weswegen bereits verschiedene Kataloge von Ideen, Prinzipien und Konzepten erstellt worden sind, die sich auf einzelne Teilbereiche oder die Informatik im Ganzen beziehen. Das Fachgebiet Datenmanagement und insbesondere dessen in den letzten Jahren hinzugekommene und neu ausgestaltete Aspekte werden jedoch bisher kaum erfasst. Dennoch lässt sich vermuten, dass auch in diesem Gebiet Konzepte mit langfristiger Relevanz existieren. Da Datenmanagement und Kompetenzen aus diesem Bereich heute auch im Alltag immer unverzichtbarer werden, sind eine didaktische Aufbereitung dieses Themenkomplexes und die Fokussierung auf grundlegende Aspekte des Gebiets notwendig. In diesem Beitrag wird daher ein Versuch unternommen, die Ermittlung der Schlüsselkonzepte eines Themenkomplexes zu systematisieren. Die vorgestellte empirische Herangehensweise wird dann auf das Fachgebiet Datenmanagement angewandt. Anhand des resultierenden Modells werden zentrale Aspekte von Datenmanagement für den Informatikunterricht exemplarisch verdeutlicht.

Keywords: Datenmanagement; Schlüsselkonzepte; Modell; Ideen; Prinzipien; Praktiken; Unterricht

1 Motivation

In der informatikdidaktischen Forschung besteht Einigkeit darüber, dass sich Unterricht nicht an möglicherweise nur kurzzeitig bedeutenden technischen Entwicklungen orientieren sollte, sondern an den grundlegenden und langfristig relevanten Aspekten des Fachs. In den letzten 20 Jahren wurden daher verschiedene Kataloge von Prinzipien, Ideen und Konzepten vorgeschlagen, die die Informatik oder einen ihrer Teilbereiche charakterisieren. Diese können beispielsweise im Rahmen der Aufbereitung neuer Themen für den Unterricht und als Grundlage der Unterrichtsplanung oder der Curriculumsentwicklung gewinnbringend eingesetzt werden. Bekannte Vertreter in der Informatikdidaktik sind beispielsweise die *Great Principles of Computing* [De03] und die *Fundamentalen Ideen der Informatik* [Sc93]. Mit der kontinuierlichen Weiterentwicklung und Ausdifferenzierung der Informatik geht jedoch oft auch eine Ergänzung oder Neuinterpretation dieser Ideenkataloge einher. So ergänzte beispielsweise Modrow die *Fundamentalen Ideen der theoretischen Informatik* [Mo03] und identifizierte einige Jahre später fundamentale Ideen der Schulinformatik [MS16].

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, andreas.grillenberger@fau.de

² Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, ralf.romeike@fau.de

Das Fachgebiet *Datenmanagement* wurde bislang noch nicht explizit auf solche Ideen hin untersucht. Gerade im Kontext von *Big Data* und damit einhergehenden Entwicklungen entstehen jedoch viele neue Themengebiete, die eigene neue Ideen mitbringen. Da diese auch Einzug in den Alltag halten, werden Kompetenzen in diesem Bereich nicht nur für Informatiker, sondern auch für den alltäglichen Umgang mit Daten immer wichtiger [GR14b]. Als Grundlage für die Aufbereitung des Fachgebiets für den Informatikunterricht ist es daher essentiell, dessen Schlüsselkonzepte zu ermitteln. Im Folgenden werden wir daher einen empirischen Ansatz zur systematischen Ermittlung vorstellen und das Modell der Schlüsselkonzepte des Datenmanagements, das wir bei der Anwendung des Ansatzes auf dieses Fachgebiet gewinnen konnten, vorstellen.

2 Datenmanagement in der Informatik und der Informatikdidaktik

In den letzten Jahren wurde der Themenbereich *Datenbanken* um eine Vielzahl neuer Themen, Methoden und Konzepte erweitert und hat sich zum Fachgebiet *Datenmanagement* weiterentwickelt. Ein zentrales Thema ist dabei *Big Data*, die Verwaltung immer größerer Datenbestände mit unterschiedlichsten Arten von Daten, die schnell verarbeitet werden müssen. Mit der Entstehung neuer Anforderungen an Datenspeicher, wie Partitionstoleranz als Basis der verteilten Datenspeicherung, und mit der Zunahme korrelationsbasierter Datenanalysen (*Data Mining*) zeigen sich Auswirkungen auf Datenbanken und andere Datenmanagementsysteme: Während beispielsweise bei klassischen Datenbanken die langfristige *konsistente Datenspeicherung* im Vordergrund steht, legen moderne *NoSQL-Datenbanken* den Fokus auf *hohe Verfügbarkeit* und *verteilte Datenspeicherung*, oft unter Vernachlässigung der Konsistenz. Gleichzeitig setzen sich neue Forschungsrichtungen im Datenmanagement insbesondere mit *Datenqualität*, *Metadaten* oder neuen Datenmanagementsystemen wie *In-Memory-Datenbanken* oder *Datenstromsystemen* auseinander.

Diese Innovationen haben unmittelbare Auswirkungen auf den Umgang mit Informatiksystemen, z. B. im Zusammenhang mit *Cloudspeichern*, *Synchronisation* persönlicher Daten über verschiedene Endgeräte hinweg, *Webdatenbanken* oder *Metadaten*. Obwohl die für den Umgang mit solchen Systemen nötigen Kompetenzen insbesondere im Informatikunterricht vermittelt werden könnten, wurden verwandte Themen in der informatikdidaktischen Forschung seit der Fundierung des Datenbankunterrichts in den 1990er Jahren kaum thematisiert. Entsprechend wurden bisher wenige Unterrichtsansätze zu moderneren Datenmanagementthemen veröffentlicht. Damit zeigt sich eine Lücke zwischen dem Stand der Forschung im Fachgebiet und dem aktuellen Schulunterricht [GR14a]: Während sich der Informatikunterricht insbesondere auf das Thema Datenbanken konzentriert, werden oben genannte Themen nur marginal betrachtet. Darüber hinaus können und sollten Schüler jedoch auch verschiedene Schlüsselkompetenzen erwerben, die für den Umgang mit Daten und Datenmanagementsystemen heute essentiell sind [GR14b]. Als praktisches Unterrichtsbeispiel haben wir das Thema Datenstromsysteme bereits in dieser Hinsicht aufbereitet [GR15a, GR15b]. Für die weitere Erschließung des Fachgebiets ist es notwendig, dieses systematisch aufzuarbeiten und die zugrunde liegenden Konzepte zu ermitteln. Da diese

einen Zugang zum gesamten Fachgebiet darstellen und ein Verständnis der zentralen Themen und Ideen ermöglichen, bezeichnen wir sie im Folgenden als *Schlüsselkonzepte*.

3 Konzepte, Prinzipien und fundamentale Ideen der Informatik

Die Strukturierung eines Fachgebiets durch Ermittlung der zugrunde liegenden Ideen, Konzepte oder Prinzipien ist insbesondere in den naturwissenschaftlichen Fächern und in der Informatik ein anerkannter Ansatz, der genutzt wird, um eine Wissenschaftsdisziplin oder einen Teil davon für den Schulunterricht zugänglich zu machen. In der Informatikdidaktik sind dabei insbesondere die *Fundamentalen Ideen der Informatik* zu nennen, die Schwill ausgehend vom Softwareentwicklungsprozess, dessen Erforschung er als eine der grundlegenden Aufgaben der Informatik erachtet, identifiziert hat [Sc93]. Dazu hat er vier (später fünf) Kriterien definiert: Eine fundamentale Idee soll einerseits in der Breite der Informatik vertreten sein („Horizontalkriterium“) und sowohl in der Vergangenheit aber auch in der erwarteten Zukunft Bedeutung haben („Zeitkriterium“). Gleichzeitig soll sie jedoch auch einen Bezug zu Denken und Alltag haben („Sinnkriterium“), auf verschiedenen kognitiven Niveaus vermittelbar sein („Vertikalkriterium“) und zum Erreichen eines idealisierten aber meist unerreichbaren Ziels dienen („Zielkriterium“). Modrow hat den Ansatz der fundamentalen Ideen aufgegriffen und auf die *Theoretische Informatik* angewendet [Mo03]. Einige Jahre später haben Modrow und Strecker den Ideenkatalog von Schwill erneut herangezogen und auf acht *fundamentale Ideen der Schulinformatik* reduziert [MS16]. Denning hingegen charakterisiert mit den *Great Principles of Computing* die Informatik aus fachlicher Sicht anhand von *Kerntechnologien*, die verschiedene Anwendungen und Forschungsrichtungen der Informatik darstellen sowie je fünf *Praktiken, Entwurfs- und Funktionsprinzipien*, die der Informatik zugrunde liegen [De03]. Für Zendler und Spannnagel waren die *Konzepte* [ZS06] und *Prozesse* [ZSK07] der Informatik zentral, die sie in einer Fragebogenstudie unter fachlichen Experten ermittelten, wobei sich die Auswahl auch an den Kriterien nach Schwill [Sc93] orientierte. Diese und weitere Ansätze unterscheiden sich in ihrer Methodik, Breite und dem Detaillierungsgrad deutlich, während sie in der Zielsetzung typischerweise übereinstimmen: Es soll die Informatik oder eines ihrer Teilgebiete durch die zugrunde liegenden Konzepte, Prinzipien bzw. fundamentalen Ideen charakterisiert werden.

4 Ermittlung der Schlüsselkonzepte des Datenmanagements

Um die Schlüsselkonzepte des Datenmanagements zu ermitteln, muss das Fachgebiet unter Beachtung der Entwicklungen der letzten Jahre aufgearbeitet und hinsichtlich der zeitstabilen Relevanz bewertet werden. Während beispielsweise Schwill und Denning ihren Katalogen eine informelle Charakterisierung der Informatik zugrunde legen, wählen wir im Folgenden einen in zwei Phasen aufgeteilten empirischen Ansatz zur systematischen Identifizierung der Schlüsselkonzepte.

4.1 Phase 1: Fachdidaktische Analyse

Im der ersten Phase soll durch Analyse zentraler fachwissenschaftlicher Literatur ein umfassender Überblick über die Themen des Fachgebiets gewonnen werden. Da dies aus fachdidaktischer Perspektive geschehen sollte, konnten bestehende Arbeiten, wie der Data Management Body of Knowledge [DA09] (DAMA-DMBoK), nicht als alleinige Basis genutzt werden: Diese beschreiben das Fachgebiet zwar ausführlich, jedoch aus fachlicher Sicht und oft stark an der professionellen Nutzung orientiert. Trotzdem können sie als Teil der Analyse wertvolle Einblicke ins Fachgebiet liefern. Unsere Analyse wurde an den Schritten der qualitativen Inhaltsanalyse nach Mayring [Ma10] orientiert. Aufgrund ihrer speziellen Zielsetzung und Aufteilung in zwei getrennte Phasen wurden jedoch einzelne im Sinne der Methodik zulässige Anpassungen vorgenommen, insbesondere durch Zusammenfassung verschiedener Schritte. Die fachdidaktische Analyse wurde daher wie folgt durchgeführt:

1. **Literaturauswahl:** Als Basis für die Analyse wurden sechs anerkannte Fachbücher [Ed11, EN09, KE15, Ku15, Pi11, UP14] aus dem Fachgebiet sowie der DAMA-DMBoK ausgewählt. Neben tradierten Grundlagenwerken wurden auch solche Werke hinzugezogen, die sich speziell auf jüngere Themen des Datenmanagements beziehen, um auch diese und damit zusammenhängende Aspekte zu berücksichtigen. Obwohl sich der Materialkanon hauptsächlich aus deutschsprachiger Literatur zusammensetzt, wurde auch die internationale Perspektive berücksichtigt: Bei einem der Lehrbücher handelt es sich um die deutschsprachige Ausgabe eines international anerkannten Fachbuchs [EN09]. Außerdem stellt der DAMA-DMBoK die Perspektive der internationalen Datenmanagement-Vereinigung dar. Im Laufe der Analyse konnten keine Abweichungen zwischen deutscher und internationaler Literatur festgestellt werden.
2. **Entscheidungen zum Kategoriensystem und den Auswahlkriterien:** Aufgrund des Ziels, einen umfassenden Überblick über das Fachgebiet zu gewinnen, ist es sinnvoll, das Kategoriensystem durch Analyse der Materialien induktiv aufzubauen. Als Analyseeinheit wurden die vorkommenden Begriffe gewählt, die Themen des Datenmanagements beschreiben. Ihr Kontext wurde dabei insofern berücksichtigt, dass auch dieser aufgenommen wurde, wenn er einen eigenen Aspekt umfasst.
3. **Analysephase:**
 - a) **Entwicklung des Kategoriensystems:** Dieser Schritt wurde in mehreren Iterationen wiederholt durchlaufen. Zuerst wurde der DAMA-DMBoK als Basis genutzt, da unsere Vermutung war, dass er das Themenfeld gut abdeckt. In den darauffolgenden Iterationen wurden zusätzliche Quellen hinzugenommen, wobei ab der dritten kaum noch weitere Themen ergänzt wurden. Stattdessen kamen eher zusätzliche Details hinzu, die als Unterkategorien aufgenommen wurden. Als Ergebnis entstand ein hierarchisch organisiertes Kategoriensystem, das zentrale Themen des Fachgebiets inklusive einiger Details, die als Subkategorien enthalten sind, beschreibt. Ein Ausschnitt der Ergebnisse wird in Abb. 1 dargestellt.

- b) **Clustering:** Nach der Entwicklung des Kategoriensystems wurde die umfangreiche Menge an Begriffen in induktiv gebildeten Gruppen reorganisiert. Dabei konnte festgestellt werden, dass sich vier Gruppen herausbilden: technische Aspekte der Funktionsweise von Datenmanagementsystemen, Prinzipien, denen Datenmanagementsysteme gehorchen, Beispiele für Anwendungen von Datenmanagement und Tätigkeiten, die bei Entwurf und Nutzung von Datenmanagementsystemen durchgeführt werden. Ein Einblick in das Clustering ist in Abb. 2 dargestellt.

Um in dieser Analysephase einen möglichst hohen Grad an Vollständigkeit sicherzustellen, wurde das Ergebnis mittels einer teilautomatisierten Textanalyse validiert. Aus einem Dokumentenkörper wurden die häufigsten Begriffe automatisch herausgefiltert und nach Themen des Datenmanagements händisch durchsucht. Als Basis dienten insbesondere Vorlesungsskripte und Folien verschiedener Wissenschaftler, die teils das gesamte Fachgebiet, teils nur einzelne Spezialgebiete abdecken, sowie weitere Lehrbücher und ein Großteil der Ausgaben des „Datenbank Spektrums“. Es wurden dafür 305 Dokumente mit 9.447 Seiten bzw. Folien analysiert, die zu über 1.800 einzuordnenden Begriffen (nach Filterung von Mehrdeutigkeiten und Stoppwörtern) führten. Die Ergebnisse wurden mit den vorherigen Ergebnissen abgeglichen, indem die häufigsten vorkommenden fehlenden Begriffe dem vorher entwickelten Kategoriensystem hinzugefügt wurden. Dabei stellten sich nur solche



Abb. 1: Ausschnitt aus den Ergebnissen des ersten Analyseschritts



Abb. 2: Clustering der zuvor gefundenen Begriffe des Datenmanagements

als fehlend heraus, die das System nicht in der Breite, sondern nur in der Tiefe erweitern. Daher kann ein hoher Vollständigkeitsgrad der Analyseergebnisse angenommen werden.

4.2 Phase 2: Ermittlung und Strukturierung der Konzepte

In der zweiten Phase wurden eine Strukturierung der Themen vorgenommen mit dem Ziel, eine prägnante aber vollständige Charakterisierung des Fachgebiets durch dessen Schlüsselkonzepte zu erhalten. Ein Vergleich der vier beim Clustering herausgebildeten Gruppen mit bereits etablierten Arbeiten zeigte, dass insbesondere das Framework der *Great Principles of Computing* diese Perspektiven abdeckt. Aufgrund der ähnlichen Zielsetzung, der Charakterisierung einer Wissenschaft bzw. eines ihrer Teilbereiche, ist eine Adaptierung des Modells für die Charakterisierung von Datenmanagement sinnvoll. Die Einordnung der zuvor gefundenen Begriffe in das Modell erfolgte daher anhand folgender auf Denning zurückzuführender Kriterien:

- **Kerntechnologien** stellen konkrete Anwendungen bzw. Technologien dar, die deutlichen Bezug zu Datenmanagement haben. Sie sind dabei gleichzeitig zentrale Forschungsrichtungen des Fachgebiets.
- **Praktiken** repräsentieren Aktivitäten und Methoden, die dem Datenmanagement zuzuordnen sind und stellen gleichzeitig Kompetenzen dar, die zur Nutzung und/oder Entwicklung von Datenmanagementsystemen notwendig sind.
- **Entwurfsprinzipien** sollten beim Entwurf von Datenmanagementsystemen berücksichtigt werden. Zusätzlich können sie auch für die Entscheidung genutzt werden, ob ein System für einen konkreten Anwendungsfall geeignet ist.
- **Funktionsprinzipien** stehen für die grundlegende Funktionsweise von Datenmanagementsystemen und sind insbesondere bei deren Implementierung relevant.



Abb. 3: Modell der Schlüsselkonzepte des Datenmanagements

5 Modell der Schlüsselkonzepte des Datenmanagements

Die beiden Phasen resultierten in einem Modell der Schlüsselkonzepte des Fachgebiets Datenmanagement (vgl. Abb. 3). In den *Kerntechnologien* spiegeln sich die Fortschritte im Datenmanagement gegenüber dem ursprünglichen Fachgebiet Datenbanken wider, aber auch zentrale Forschungsrichtungen des Fachgebiets: Dies sind derzeit Dateispeicher, Datenbanken, Datenstromsysteme, Datenanalysen, Semantic Web und Dokumentenspeicher. Wie bei Denning ist zu erwarten, dass zukünftig weitere Kerntechnologien hinzukommen.

Die *Praktiken* entsprechen den Aktivitäten und Methoden, die im Kontext von Datenmanagement notwendig sind. Dies schließt die Gewinnung und Bereinigung von Daten, die Datenmodellierung, Implementierung und die Optimierung, Analyse und Visualisierung von Daten sowie die Evaluation der Analyseergebnisse, aber auch den Austausch und schlussendlich die Archivierung bzw. Löschung von Daten mit ein.

Bei der Entwicklung und Nutzung von Datenmanagementsystemen müssen die folgenden *Entwurfsprinzipien* beachtet werden:

- **Datenunabhängigkeit**, d. h. die Abstraktion zwischen interner Repräsentation von Daten und ihrer Schnittstelle nach außen.
- Sicherstellung der **Integrität** und der **Konsistenz** von Datenbeständen.
- **Isolierung** und **Dauerhaftigkeit** von Änderungen im Datenbestand.
- **Verfügbarkeit** von Daten, d. h. geeignete und schnelle Zugriffsmöglichkeiten.
- **Partitionstoleranz**, d. h. Toleranz gegenüber Ausfällen von Teilen eines verteilten Datenspeichers.
- **Konkurrenz** von Anfragen in Datenmanagementsystemen und Datenanalyseprozessen bei nebenläufigem Zugriff.
- (Vermeidung oder Nutzung von) **Redundanz** in Datenbeständen.

Die grundlegende Funktionsweise von Datenmanagementsystemen basiert auf folgenden *Funktionsprinzipien*:

- **Strukturierung** von Daten (z. B. durch Metadaten oder Indizes), um auf diese schnell und effizient zugreifen zu können.
- **Repräsentation**, d. h. Methoden und Techniken zur Speicherung von Daten, beispielsweise (interne) Datenstrukturen.
- **Replikation** und **Partitionierung**, d. h. mehrfache und/oder verteilte Speicherung von Daten über mehrere Datenspeicher hinweg.
- **Synchronisation** von gleichzeitigen bzw. konkurrierenden Zugriffen auf Daten.
- **Transport** von Daten, der die Übertragung von Daten, aber auch die Übertragungssicherheit einschließt.
- **Transaktionen**, die insbesondere zur Erhöhung der Fehlertoleranz eingesetzt werden.

6 Anwendung und Relevanz für den Informatikunterricht

Das Modell der Schlüsselkonzepte des Datenmanagements liefert einen umfangreichen Überblick über dessen zentrale Konzepte aus Sicht relevanter Technologien, Praktiken und Prinzipien. Es ist klar erkennbar, dass das Fachgebiet eine große Vielfalt an Themen umfasst, die im Unterricht bisher allenfalls am Rande betrachtet werden. Trotzdem haben diese Themen und die damit in Verbindung stehenden Kompetenzen im Zusammenhang mit der zunehmenden Bedeutung von Datenmanagement auch im Alltag eine hohe Relevanz für den Informatikunterricht. Das Modell eignet sich damit insbesondere, um bei der Unterrichts- oder Curriculumsentwicklung zu untersuchen, welche Aspekte des Fachgebiets durch ein Unterrichtsthema abgedeckt werden können bzw. sollen. Exemplarisch kann hier das Thema Datenstromsysteme [GR15a] herangezogen werden, das auch schon außerhalb der Informatik und Informatikdidaktik zunehmend Aufmerksamkeit erhält (vgl. [Tu16]): Datenstromsysteme werden zur Echtzeitverarbeitung immer größerer Datenmengen eingesetzt und machen deutlich, wie sich die Grenzen im Datenmanagement gegenüber den vor einigen Jahren noch relativ zeitaufwändigen Datenanalysen verschoben haben. Dieses Thema umfasst insbesondere folgende Schlüsselkonzepte:

Kerntechnologie:	Datenstromsysteme
Praktiken:	Datenerfassung/-gewinnung, Datenbereinigung, Analyse
Entwurfsprinzipien:	Verfügbarkeit, Konkurrenz
Funktionsprinzipien:	Strukturierung, Repräsentation, Transport

Die Betrachtung des Themas Datenstromsysteme aus der Perspektive des Modells der Schlüsselkonzepte von Datenmanagement hilft, die zentralen Konzepte des Themas zu erkennen und im Unterricht einen Schwerpunkt auf diese zu legen.

Im Vergleich mit dem zentralen Thema „Datenbanken“ im aktuellen Informatikunterricht zum Thema Daten ist die unterschiedliche Ausprägung der Schlüsselkonzepte des Datenmanagements in beiden Systemen klar erkennbar:

Kerntechnologie:	Datenbanksysteme
Praktiken:	Modellierung, Implementierung, Optimierung, Analyse
Entwurfsprinzipien:	Datenunabhängigkeit, Integrität, Konsistenz, Redundanz
Funktionsprinzipien:	Strukturierung, Repräsentation, Transaktion

Während bei Datenstromsystemen die Echtzeitdatenverarbeitung zentral ist, setzen Datenbanken den Fokus auf strukturierte Speicherung. Beide vorgestellte Themen sind exemplarisch für das Fachgebiet und verdeutlichen, dass verschiedene Schlüsselkonzepte sich als rote Fäden durch dieses ziehen, aber unterschiedlich ausgeprägt sind. Das Modell kann damit zu einer Fokussierung auf die zentralen Aspekte beitragen, liefert einen Überblick über zentrale Begriffe des Fachgebiets und kann als Zugang zu diesem herangezogen werden.

7 Fazit

In diesem Beitrag wurde eine empirische Vorgehensweise zur systematischen Ermittlung der Schlüsselkonzepte eines Fachgebiets vorgestellt. Durch Anwendung auf das Fachgebiet Datenmanagement konnten wir ein Modell der Schlüsselkonzepte dieses Fachgebiets erstellen, das durch die vier Kategorien Kerntechnologien, Praktiken, Entwurfs- und Funktionsprinzipien die zentralen Aspekte des Fachgebiets betont. Vergleicht man dieses Modell mit den *Great Principles of Computing* [De03], so zeigt sich, dass es an verschiedenen Stellen eine Spezialisierung der Great Principles darstellt: Beispielsweise wird das von Denning genannte Entwurfsprinzip *Sicherheit* im Datenmanagement durch die Entwurfsprinzipien *Integrität*, *Verfügbarkeit* und *Isolierung* konkretisiert. Betrachtet man die Kriterien, die Schwill an fundamentale Ideen der Informatik stellt [Sc93], lässt sich vermuten, dass die ermittelten Schlüsselkonzepte repräsentativ für fundamentale Ideen stehen: Einerseits sind sie in verschiedenen Bereichen des Fachgebiets, aber häufig auch darüber hinaus, relevant und erfüllen somit das Horizontalkriterium. Ebenso scheint auch das Vertikalkriterium erfüllbar, wie beispielsweise durch die Aufarbeitung des Themas Datenstromsysteme als Unterrichtsbeispiel [GR15a] gezeigt wurde. Durch den Alltagsbezug der Konzepte wird auch das Sinnkriterium adressiert, durch die langjährige Relevanz der meisten Aspekte auch das Zeitkriterium. Trotzdem unterscheiden sich die dargestellten Schlüsselkonzepte von den Ideen: Während letztere die Urbilder bezeichnen, aus denen informatische Phänomene entstehen, sind die Konzepte auf einer konkreteren Ebene angesiedelt und beschreiben Gesetzmäßigkeiten, Funktionsweisen und Vorgehensweisen im Datenmanagement.

Während sich Informatikunterricht im Bereich „Daten“ bisher auf Datenbanken konzentriert, verdeutlichen die Schlüsselkonzepte die Vielfalt von Datenmanagement. Durch die strukturierte Darstellung kann das Modell einen deutlichen Beitrag zur Curriculums- und Unterrichtsgestaltung liefern, und als Basis für die weitere Forschung eingesetzt werden. Wie exemplarisch gezeigt wurde, eignet sich das Modell, um Themen des Datenmanagements hinsichtlich der enthaltenen Konzepte zu untersuchen. Für die Schule ergeben sich verschiedene Vorteile: Zum einen wird das Fachgebiet strukturiert und der Kompetenzerwerb konkretisiert, zum anderen liefern die Schlüsselkonzepte konkrete Begriffe, die den fundierten Gebrauch der Fachsprache unterstützen. Die vorgestellte Herangehensweise

zur Ermittlung der Schlüsselkonzepte konnte damit am Fachgebiet Datenmanagement erfolgreich angewandt werden und scheint auch bezogen auf andere Gebiete der Informatik und möglicherweise darüber hinaus erfolgversprechend.

Literaturverzeichnis

- [DA09] DAMA International: The DAMA Guide to the Data Management Body of Knowledge - DAMA-DMBOK. Technics Publications, LLC, USA, 2009.
- [De03] Denning, P. J.: Great Principles of Computing. Commun. ACM, 46(11):15–20, 2003.
- [Ed11] Edlich, S. et al.: NoSQL. Hanser Fachbuchverlag, 2011.
- [EN09] Elmasri, R. A.; Navathe, S. B.: Grundlagen von Datenbanksystemen. Pearson Studium, 2009.
- [GR14a] Grillenberger, A.; Romeike, R.: A Comparison of the Field Data Management and its Representation in Secondary CS Curricula. In: Proceedings of WiPSCE 2014. ACM, Berlin, 2014.
- [GR14b] Grillenberger, A.; Romeike, R.: Teaching Data Management: Key Competencies and Opportunities. In: Proceedings of KEYCIT 2014. Universitätsverlag Potsdam, 2014.
- [GR15a] Grillenberger, A.; Romeike, R.: Big-Data-Analyse im Informatikunterricht mit Datenstromsystemen: Ein Unterrichtsbeispiel. In: Praxisband INFOS 2015. 2015.
- [GR15b] Grillenberger, A.; Romeike, R.: Big Data im Informatikunterricht: Motivation und Umsetzung. In: INFOS 2015. Lecture Notes in Informatics (LNI). Köllen, Bonn, 2015.
- [KE15] Kemper, A.; Eickler, A.: Datenbanksysteme. Gruyter, Walter de GmbH, 2015.
- [Ku15] Kudraß, T.: Taschenbuch Datenbanken. Hanser Fachbuchverlag, 2015.
- [Ma10] Mayring, P.: Qualitative Inhaltsanalyse: Grundlagen und Techniken. Beltz, 2010.
- [Mo03] Modrow, E.: Fundamentale Ideen der theoretischen Informatik. In: Informatische Fachkonzepte im Unterricht, INFOS 2003. S. 189–200, 2003.
- [MS16] Modrow, E.; Strecker, K.: Didaktik der Informatik. Oldenbourg, 2016.
- [Pi11] Piepmeyer, L.: Grundkurs Datenbanksysteme. Hanser Fachbuchverlag, 2011.
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik, 25(1), 1993.
- [Tu16] Tulodziecki, G.: Konkurrenz oder Kooperation? Zur Entwicklung des Verhältnisses von Medienbildung und informatischer Bildung. MedienPädagogik: Zeitschrift für Theorie und Praxis der Medienbildung, 25(0), 2016.
- [UP14] Unland, R.; Pernul, G.: Datenbanken im Einsatz. de Gruyter, Oldenbourg, 2014.
- [ZS06] Zendler, A.; Spannagel, C.: Zentrale Konzepte im Informatikunterricht: eine empirische Grundlegung. Notes on Educational Informatics—Section A: Concepts and Techniques, 2(1):1–21, 2006.
- [ZSK07] Zendler, A.; Spannagel, C.; Klautd, D.: Zentrale Prozesse im Informatikunterricht: eine empirische Grundlegung. Notes on Educational Informatics—Section A: Concepts and Techniques, 3(1):1–19, 2007.

Entwicklung eines Datenschutzkompetenzmodells

Alexander Hug¹, Rüdiger Grimm²

Abstract: Zur Beschreibung der Fähigkeit im Umgang mit (digitalen) Medien haben Six und Gimmler ein Medienkompetenzmodell veröffentlicht [SGG07]. Darin fehlen Punkte wie Risikobewertung und Vermeidungsstrategie, die unter dem Gesichtspunkt des Datenschutzes besonders zu beachten sind. Risikobewertungen können nach den Regeln der IT-Sicherheit vorgenommen und mithilfe des Vertrauensmodells von Mayer, Davis und Schoorman [MDS95] explizit gemacht werden. In diesem Artikel wird das Medienkompetenzmodell zu einem begründeten Datenschutzkompetenzmodell erweitert. Dabei erfolgt die Risikobewertung über ausgewählte Dimensionen des Medienkompetenzmodells. Ein derart erweitertes Modell liefert eine fachdidaktisch begründete Vorlage zur Messung der Risikobewertung bei Schülerinnen und Schülern in Bezug auf die Wahrung ihrer Privatsphäre.

Keywords: Datenschutz, Datenschutzkompetenzmodell, Medienkompetenzmodell, Vertrauensmodell, Referenzmodell IT-Sicherheitsanalyse

1 Internetnutzung als ein Zusammenspiel zwischen Selbstkontrolle und Vertrauen

Die Nutzung des Internets ist ohne eine Kooperation mit verschiedenen Parteien (Softwareherstellern, Providern, Diensteanbietern), denen man Vertrauen entgegenbringen muss, nicht möglich. In [GB15] wird Vertrauen definiert „als eine Bereitschaft des Trustors [des Vertrauensnehmers], eine riskante Handlung in einem Kontext zu unternehmen, die er nicht vollständig kontrolliert, in der Erwartung, dass der Trustee [der Vertrauensgeber] diesen kontrolliert und den Trustor darin schützt.“

Das Zusammenspiel zwischen Trustor und Trustee wird durch das Modell von Mayer, Davis und Schoorman beschrieben [MDS95]. Der Nutzer (Trustor) schenkt den Anbietern (Trustees) aufgrund deren Kompetenz, deren Wohlwollen und deren Integrität sein Vertrauen. Liegt nun ein zu erwartendes Risiko vor, dann wird dieses in die Vertrauensbeziehung mit aufgenommen. Aufgrund der Wirkung der Ergebnisse dieser Vertrauensbeziehung wächst oder fällt mit der Zeit das Vertrauen, da die Ergebnisse wiederum die wahrgenommene Vertrauenswürdigkeit des Trustees darstellen. Dies stellt ein rückkoppelndes Element dar.

Handlungen, die im Zusammenhang mit der Internetnutzung vollzogen werden, sind risikobehaftet. Bevor der Nutzer den Trustees Vertrauen schenkt, muss er das Risiko

¹ Universität Koblenz-Landau, FB Informatik, Universitätsstr. 1, 56070 Koblenz, hug@uni-koblenz.de

² Universität Koblenz-Landau, FB Informatik, Universitätsstr. 1, 56070 Koblenz, grimm@uni-koblenz.de

wahrnehmen und bewerten. Mit Hilfe des Referenzmodells für ein Vorgehen bei der IT-Sicherheitsanalyse [Gr16] kann eine Bewertung vorgenommen werden. Es ist in erster Linie für die Entwickler von Sicherheitssystemen gedacht und kann in abgewandelter Form auch zur Einschätzung eines Nutzungsrisikos genutzt werden. Bei dieser Abwandlung werden nur die datenschutzrelevanten Aspekte und Kriterien des vierschrittigen Modells benutzt.

Im ersten Schritt, der sog. Ist-Analyse, sind die Güter in diesem Fall die persönlichen Daten. Als Akteure gelten der Nutzer, die Administratoren der Netzwerke, die Dienstleister und letztendlich auch die Personen, die unbefugt in den Prozess des Datenaustauschs eingreifen. Interessenkonflikte können in dem, was der Nutzer erwartet, und dem, was sich die anderen Kommunikationsteilnehmer wünschen, entstehen. Schwachstellen im System können durch eine falsche Konfiguration der Hard- und Software, durch Fehler in der Software, aber auch durch Missachtung von Verhaltensregeln entstehen.

Der zweite Schritt ist die Potenzial-Analyse. Bedrohungen als mögliche Folge von Angriffen (z. B. auf das Endgerät des Nutzers und auf den Kommunikationsprozess) sind z. B. eine Fremdsteuerung des eigenen Geräts, Datenverlust (etwa durch Diebstahl von Kreditkartendaten), das unbefugte Mitlesen von Kommunikation oder eine zweckfremde Nutzung von veröffentlichten Daten in sozialen Netzwerken. Dies bedeutet im letzten Fall konkret, dass aufgrund aggregierter Daten aus dem Profil, der Zeitleiste, den Chat-Verläufen, usw. entweder ein falsches Profil des Nutzers entstehen kann oder das Profil stimmig ist, aber der Nutzer aufgrund anschließender Einflussnahme des Betreibers unerwünschte Werbung, Spam-Mails o. Ä. erhält. Das Risiko ist daher immer in Bezug auf die schützenswerten Güter zu bemessen. Im Falle der personenbezogenen Daten ist ihr Bedrohungsrisiko je nach Datenart und nicht immer direkt monetär zu beziffern. Schwachstellen, die hier ausgenutzt werden können, sind zu leichte Zugänge zu den Servern (z. B. durch schwache Passwörter), unzureichend geschützte Zugänge zu den Daten (offene Netze) und das Nutzerverhalten (z. B. der Verzicht kryptografischer Verfahren). Die Anforderung eines Schutzes an Privatheit lässt sich so in folgende funktionale Sicherheitsanforderungen zerlegen: Vertraulichkeit und Zweckbindung der Daten, sowie Vertrauenswürdigkeit der anderen Parteien und eine funktionsintegre Verfügbarkeit der Daten.

Im nächsten Schritt wird ein Sicherheitskonzept entwickelt. Eine erste Sicherheitsmaßnahme ist die Nutzung von starken Passwörtern, von Verschlüsselung beim Mailen, Chatten und im Web. Ferner sollte der Nutzer durch Kenntnis der Datenschutzprinzipien darauf achten, dass er nur ausgewählte Inhalte kommuniziert und diese mit Bedacht weitergibt. Weiterhin sind Sicherheitseinstellungen vorzunehmen und Schutzmechanismen zu nutzen, etwa alternative Suchmaschinen oder das regelmäßige Löschen von Cookies und des Browserverlaufs.

Der letzte Schritt ist die Installation eines Sicherheitskonzepts. Es dient dem Internetnutzer, diejenigen Sicherheitsmechanismen des Selbst Datenschutzes

auszuwählen, die ihm ein adäquates Sicherheitsniveau bieten. Dabei muss er das verbleibende Restrisiko erkennen, akzeptieren und entscheiden, inwieweit er den Partnern, von denen er dann noch abhängt, vertraut. Hier ist die Handlungsfähigkeit in der Datenschutzkompetenz gefragt.

2 Von der Risikobewertung in einem Medienkompetenzmodell zu einem Datenschutzkompetenzmodell

In [SGG07] stellen Six und Gimmler ein Medienkompetenzmodell vor, das – wie Gimmler in [Gi12] schon zeigt – Datenschutzkompetenz in einem gewissen Maß zu beschreiben vermag. Aber eine Betrachtung des Risikos bei der Internetnutzung unterbleibt an dieser Stelle. Folgende Dimensionen aus dem Medienkompetenzmodell spielen bei der Risikobewertung eine Rolle:

Hintergrundwissen ist notwendig, da zur Abschätzung des Risikos der Internetnutzung die möglichen Schadensursachen und ihrer Gegenmittel ja bekannt sein müssen. Die Nutzung alternativer Softwareprodukte, z. B. alternativer Suchmaschinen und Verschlüsselungsmethoden, und die eigene Entscheidung über die Datenpreisgabe sind Datenschutzmaßnahmen, die der Nutzer ergreifen kann, wenn er sie kennt. Ferner zählen wir zu dieser Dimension auch das Wissen über die Prinzipien des Datenschutzes und deren Bedeutung und Anwendung.

Der Selbstschutz verlangt Orientierungswissen, weil der Nutzer die Funktionen der Angebote verstehen muss. Er muss in der Lage sein, diese Funktionen in der Anwendungsumgebung, in der er sich befindet, sicher und wirkungsgemäß auszuführen.

Der Nutzer kennt die Angebote, kann die angemessenen auswählen und setzt sie aus dieser Orientierungssicherheit heraus situationsgerecht ein. Ferner kennt er Möglichkeiten, um sich weitere und neue Informationen zu erschließen.

Zur Urteilskompetenz zählt die Fähigkeit, die Angebote im Internet zu kennen und deren Wirkung zu beurteilen. Zusätzlich muss sich der Nutzer bei der Abwägung des Risikos ein Urteil über die Vertrauenswürdigkeit der anderen Kommunikationsteilnehmer bilden. Auf der Basis der Urteilsentscheidung nimmt der Nutzer das Angebot an oder lehnt es ab.

Durch die Auswahl- und Nutzungskompetenz kommt der Nutzer zu der „selbstbestimmten, zielorientierten und reflektierten Auswahl und Nutzung“ [Gi12] des Internetangebots. In risikobehafteten Situationen urteilt der Nutzer, ob er das Risiko eingehen wird oder nicht. Durch die passende Auswahl und Nutzung von Schutzmechanismen kann der Nutzer das Risiko minimieren.

Die Dimensionen des Medienkompetenzmodells sind nicht völlig unabhängig voneinander (bspw. Aspekte von Urteil und Orientierung), daher ist es hilfreich, einzelne Kompetenzaspekte mehrfach zuzuordnen. Die auf die ausgewählten Dimensionen des

Medienkompetenzmodells angewendete Risikointerpretation führt zu einer Erweiterung des Modells, welches als Datenschutzkompetenzmodell aufgefasst werden kann. Damit definieren wir Datenschutzkompetenz als den Zusammenschluss von Hintergrundwissen, Orientierungswissen, Urteilskompetenz, Handlungs- und Nutzungskompetenz, Risikobewertungskompetenz und die Anwendung von Handlungsmustern mit Bezug auf das schätzenswerte Gut der persönlichen Daten.

3 Zusammenfassung und Ausblick

Eine datenschutzbewusste Internetnutzung ist gekennzeichnet durch die bewusste Ausübung des Selbst Datenschutzes im Rahmen des wahrgenommenen bestehenden System Datenschutzes und seiner Vertrauenswürdigkeit bei gleichzeitig kompetenter Abwägung des Restrisikos. Aufgrund des Medienkompetenzmodells von Six/Gimmler [SGG07], das Teilkompetenzen bzgl. des Datenschutzes aufzeigt, aber Risikobewertung außer Acht lässt, wurde das Modell zu einem Datenschutzkompetenzmodell erweitert. Die Validität dieses Modells soll im Rahmen einer weiteren Forschung untersucht werden. Auf Basis der Modellvorlage wird eine Schülerbefragung mit dem Schutz der Privatsphäre als Schwerpunkt durchgeführt, sodass aufgrund dieser Ergebnisse später konkrete Handlungsempfehlungen für Unterrichtsinhalte entwickelt werden.

4 Literaturverzeichnis

- [GB15] Grimm, R.; Bräunlich, K.: Vertrauen und Privatheit. Anwendung des Referenzmodells für Vertrauen auf die Prinzipien des Datenschutzes. In Datenschutz und Datensicherheit - DuD, 2015, 39; S. 289–294.
- [Gi12] Gimmler, R.: Medienkompetenz und Datenschutzkompetenz in der Schule. In Datenschutz und Datensicherheit - DuD, 2012, 36; S. 110–116.
- [Gr16] Grimm, R. et al.: Referenzmodell für ein Vorgehen bei der IT-Sicherheitsanalyse. In Informatik-Spektrum, 2016, 39; S. 2–20.
- [MDS95] Mayer, R. C.; Davis, J. H.; Schoorman, F. D.: An Integrative Model of Organizational Trust. In Academy of Management Review, 1995, 20; S. 709–734.
- [SGG07] Six, U.; Gleich, U.; Gimmler, R. Hrsg.: Kommunikationspsychologie -- Medienpsychologie. Lehrbuch. BeltzPVU, Weinheim, 2007.

Vom Flaggenalphabet zur Vorratsdatenspeicherung: Schülerinnen und Schüler als Multiplikatoren technischer Aspekte der digitalen Welt

Klaus-Tycho Förster¹

Abstract: Im Nachrichtenalltag existieren viele kontroverse informatische Phänomene wie etwa die Gesundheitskarte, Internetsperren oder die Vorratsdatenspeicherung. Bei Berichten oder Diskussionen über die Thematiken fällt zumeist auf, dass zwar über den informatischen Kontext gesprochen, das zur Beurteilung benötigte technische Wissen jedoch eher ignoriert, teilweise sogar sachlich falsch verwendet wird. Durch informatisch-technische Bildung erhalten Schülerinnen und Schüler die Möglichkeit, zumindest in ihrem lokalen Umfeld als Multiplikatoren dieser Aspekte zu wirken. Die Verbindung von Informatik und Gesellschaft unter technischen Aspekten im Informatikunterricht ist dabei natürlich kein neuartiges Konzept. Dennoch möchte die vorgestellte mehrfach praktisch im Schulalltag erprobte Unterrichtssequenz einen kurzen Vorschlag liefern, wie dieses für die Thematik der Vorratsdatenspeicherung im Rahmen der Behandlung von Computernetzen durchgeführt werden könnte.

Keywords: Computernetzwerke, Vorratsdatenspeicherung, technische Aspekte der Informatik

1 Schülerinnen und Schüler als lokale Multiplikatoren

Die Idee der kontextorientierten Unterrichtsorganisation ist ein *neues altes Unterrichtsparadigma* (vergl. [Ko]) zur Herstellung des Lebensweltbezuges im Informatikunterricht. Unter dem Oberbegriff *Informatik im Kontext (IniK)* werden dabei diverse Thematiken aufgegriffen, siehe dazu etwa etwa [DKW11] und [DD11]. Der Lebensweltkontext soll dabei nach Diethelm, Koubek und Witten auch das Kriterium erfüllen, dass er geeignet für „*Abendbrotgespräche mit den Eltern ist*“ [DKW11]. Daher bietet sich eine Multiplikatoren-Rolle der Schülerinnen und Schüler aus dem Informatikunterricht heraus an.

Bei anderen schulischen fächern zuordenbaren Phänomenen, wie z.B. der Nuklearkatastrophen von Fukushima und Tschernobyl, haben Eltern zumeist aus ihrer eigenen Lebenserfahrung bzw. ihrer (Schul-)Bildung einen Wissensvorsprung gegenüber ihren Kindern in der Sekundarstufe, der neben den gesellschaftlichen auch die technischen Aspekte umfasst. In Bezug auf technische Aspekte der Informatik bei tagesaktuellen Phänomenen dreht sich diese althergebrachte Rollenverteilung jedoch oft um.

¹ Aalborg University, Department of Computer Science, Selma Lagerlöfs Vej 300, DK-9220 Aalborg, Denmark, ktfoster@cs.aau.dk. Klaus-Tycho Foerster is supported by the Danish Villum Foundation.

Sofern die Eltern nicht ein erhebliches fachliches Interesse an der Informatik zeigen, begegnet ihnen der technische Hintergrund nur in alltäglichen Zusammenhängen, wie etwa in Nachrichten, Zeitungsartikeln oder Diskussionssendungen in Radio und Fernsehen. Dabei ist ohne Vorkenntnisse nur schwer zu überprüfen, ob geführte Argumentationsketten oder (anscheinend) fachliche Aussagen auch der Realität entsprechen. Die gesellschaftlichen Implikationen sind nachvollziehbar, basieren aber gegebenenfalls auf verfälschten informatischen Annahmen.

Auf Grund der teilweisen Rollenkehr ist es rein mit technischen Argumenten für Schülerinnen und Schüler der Sekundarstufe I eher schwierig, mit ihren Eltern gemeinsam über informatische Phänomene zu diskutieren. Auch bei Kenntnis der Positionen diverser gesellschaftlicher Gruppierungen und Interessenverbänden liegt eine Akzeptanz der entgegengesetzten Meinung durch die Schülerinnen und Schüler nicht unbedingt vor, dieser gegenseitige Respekt ist für eine Diskussion jedoch unabdingbar: Teilweise kommt es zu der Kollision von technischem Optimismus und gesellschaftlichem Pessimismus. Ein aktives Erleben und Nachvollziehen der Meinungsbildung scheint daher unerlässlich für eine aktive Multiplikatoren-Rolle der Schülerinnen und Schüler gegenüber ihrem lokalen Umfeld.

In anderen Schulfächern, wie etwa Sozialkunde oder Politik, wird dazu im Vergleich zur Informatik häufiger die Methode der Podiumsdiskussion eingesetzt. Anhand durch die Lehrkraft vorbereitetes oder durch die Schülerinnen und Schüler gesuchtes Quellenmaterial werden Positionen vorher definierter Gruppen ausgearbeitet, die in einem Rollenspiel ähnlich einer Talkshow ausdiskutiert werden. Dabei wird auch das Publikum aktiv eingebunden: *„Diese Art von Rückkopplung dient dazu, die Klassen stärker in die Diskussion miteinzubeziehen und den Realitätsbezug der Positionen zu erhöhen“* [Bu04].

Natürlich wird die Methode des Rollenspiels auch im Informatikunterricht schon seit längerem verwendet, z.B. zur Simulation einer CPU [Gi] oder der Einführung der objektorientierten Modellierung [Di07, S.86ff]. Für eine weitergehende Betrachtung der Thematik siehe etwa [Fo07]. Die Methode des Rollenspiels scheint jedoch in der Fachliteratur zumeist auf eher technische bzw. algorithmische Aspekte der Informatik angewandt zu werden.

2 Vorratsdatenspeicherung im technischen Bezug von Computernetzen

Es stellt sich dabei die Frage, warum diese Multiplikatoren-Rolle für die Vorratsdatenspeicherung insgesamt in einem Informatikunterricht behandelt werden muss. Man könnte meinen, dass man die physikalischen Grundlagen der Nachrichtenübermittlung auch in Physik, nur die benötigten Netzelemente in Informatik und dann abschließend davon getrennt die gesellschaftlichen Auswirkungen in einer Podiumsdiskussion im Politik- bzw. Sozialkundeunterricht vermitteln kann. Insbesondere in Bezug auf die gesellschaftlichen Aspekte lässt sich diese Auslagerung in andere Fächer nach Schubert und Schwill jedoch nicht vornehmen:

„Ethische und rechtliche Regelungen spielen in diesem Unterrichtsgegenstand eine große Rolle. Wenn sich Informatiklehrer dieser Aspekte nicht annehmen, gehen die fachlichen Bezüge im Lernprozess verloren. Man kann sicher in anderen Fächern darüber sprechen, aber kaum Verständnis für die komplexen Zusammenhänge entwickeln. Die Wirkungsprinzipien der Internet-Architektur bilden die kognitive Basis für das Verstehen der komplizierten Umsetzung der Gesetzgebung.“ [SS04, S.267f.].

Ein eigenes Erfahren und Erforschen durch die Schüler ist dabei sicherlich sinnvoll, *„weil die Lernenden eher an den Werkzeugen und Anwendungsbeispielen, vor allem aber am eigenen Tun interessiert sind als an den Fachinhalten“* [Mo03, S.189].

Die Schülerinnen und Schüler für die technische Beherrschung und gesellschaftliche Beurteilung von Computernetzen zu befähigen, wird bei der Konzeption von Informatikunterricht in Zukunft sicherlich eine immer wichtiger werdende Rolle spielen, denn die Frage *„Wie kommunizieren zwei Rechner miteinander?“* ist nach Aussage von Hubwieser von hoher allgemein bildender Bedeutung [Hu07, S.245]. Schubert und Schwill verweisen ebenfalls darauf, dass im Falle der Informations- und Kommunikationssysteme der Bildungswert sehr hoch eingeschätzt werden muss, da diese Systeme unter anderem folgendes Ziel für Schüler ermöglichen: *„Sie verstehen die Wirkprinzipien von Rechnernetzen und verteilten Systemen, der Basistechnologie neuer gesellschaftlicher Entwicklungen“* [SS04, S.255]. Es sollte daher während der schulischen Ausbildung rechtzeitig und regelmäßig bei den Schülerinnen und Schülern gesorgt werden für die

- §3.1 Präzisierung des Begriffes der Kommunikation und Analyse ihrer Komponenten
- §3.2 Entwicklung eines technischen Grundverständnisses über Aufbau und Funktion von Computernetzen
- §3.3 Fähigkeit zur sachgerechten Beurteilung der Vorteile und Risiken der Kommunikation in Computernetzen

Aufbauend auf diesen Kompetenzen kann eine gesellschaftliche Einordnung des Phänomens der Vorratsdatenspeicherung unter Berücksichtigung technischer Aspekte (§3.4) im Informatikunterricht durch die Schülerinnen und Schüler im Rahmen etwa einer Podiumsdiskussion geleistet werden. Ohne die vorherige technische Grundlagenlegung in (§3.1 - §3.3) kann die komplexe Thematik der Vorratsdatenspeicherung von den Schülerinnen und Schülern nicht adäquat beurteilt werden, sie sind ein notwendiger Baustein.

Ein kurzer Vorschlag für eine mögliche Umsetzung der Thematik wird dazu im nächsten Abschnitt gegeben. Er basiert auf einer mehrfach durchgeführten Unterrichtssequenz im Umfang von ca. 4 Doppelstunden, welche im Informatikunterricht der 9. Klasse gehalten wurde. Die Doppelstundenthemen ergeben sich aus dem obigen Aufbau:

- §3.1 Wie funktioniert Kommunikation?
- §3.2 Wesentliche Aspekte eines Computernetzes
- §3.3 Untersuchung von Eigenschaften eines Computernetzes
- §3.4 Auswirkungen der Vorratsdatenspeicherung

Natürliche Erweiterungen ergeben sich durch ausführlichere Behandlung der Anonymität in Computernetzen und der Planung von Computernetzen für diverse Anwendungsfälle. Weiteres Beispielmateriale und vorläufige Ausführungen finden sich in [Fo10].

3 Ein praktischer Durchführungsvorschlag

3.1 Wie funktioniert Kommunikation?

- Welche Kommunikationsarten gibt es, welche Regeln gelten hierbei?
- „Analoge“ Beispiele: Morsen, Winkieralphabet mit Flaggen, Seefunk
- Welche Besonderheiten gibt es bei Nachrichten zwischen Computern?
- Wie sehen Pakete im Netzwerk vereinfacht aus?

Die Grundlagen von Computernetzen basieren auf der Kommunikation mit Protokollen, während die Vorratsdatenspeicherung Protokolle über die Kommunikation anlegt. Ein Einstieg in die Unterrichtssequenz über die Vorratsdatenspeicherung wäre zwar möglich, für Schülerinnen und Schüler der Sekundarstufe I scheint die Thematik zu diesem Zeitpunkt jedoch noch zu abstrakt zu sein, bzw. führt nicht direkt zu den dahinterliegenden technischen Aspekten. Nach eigener Erfahrung reicht die Behandlung von Computernetzen anscheinend als intrinsische Motivation in der 9. Klassenstufe aus, jedoch nicht für einen Einstieg über die Eigenschaften von Kommunikation – hier bietet sich eine aktive Erforschung durch die Schülerinnen und Schüler an.

Die praktische Untersuchung nur einer einzelnen Kommunikationsart würde jedoch wie bei der Verwendung nur einer einzelnen Programmiersprache (vergl. [St11]) eher zu Produktwissen führen, statt allgemeingültige Komponenten wie Sender und Empfänger zu identifizieren. Für viel Freude bei den Schülerinnen und Schülern sorgt dabei das praktische Ausprobieren in Gruppen von Morsen mit Taschenlampen und dem Einsatz von je zwei Signalflaggen pro Sender. Abhängig vom Zeitansatz können weitere Kommunikationsarten hinzugefügt werden (z.B. Zeichensprache) bzw. Hilfsmittel wie eine Morsecode-Tabelle weggelassen werden. Eine Vertiefung dieser Thematik für weitere Unterrichtsstunden in der Oberstufe wird in [Me12] ausgeführt. Ausgehend von diesem Grundverständnis der Kommunikation kann in der verbleibenden Zeit der ersten Doppelstunde eine stark vereinfachte Version der paketbasierten Nachrichtenübermittlung zwischen Computern durch die Schülerinnen und Schüler erarbeitet werden.

3.2 Wesentliche Aspekte eines Computernetzes

- (Gruppenpuzzle) Was ist bzw. wie funktioniert es?
- (Stammgruppe) eine MAC- bzw. IP-Adresse
- (Expertengruppen) Router, Switch, WLAN?

Hartman et al. [HNR06] formulieren als eine zentrale Leitidee, dass die Adressierung ein wichtiger Aspekt bei Netzprotokollen sei und eine Grundvoraussetzung für die funktionierende Kommunikation. Explizit wird dabei erwähnt, dass die Lernenden auch in

der Lage sein sollen, selbständig MAC- und IP-Adresse herauszufinden [HNR06, S.54f]. Daher werden zunächst die notwendigen technischen Fachkenntnisse aufgebaut, bevor in der nächsten Doppelstunde das Netz an sich erforscht wird.

Durch die Vielfältigkeit des Themas wird eine Untersuchung der gängigen Komponenten eines Computernetzwerkes mithilfe eines Gruppenpuzzles durchgeführt. Nach einer Betrachtung von MAC und IP-Adressen in Stammgruppen werden in Expertengruppen die Bereiche Router, Hub/Switch und WLAN bearbeitet, um anschließend in den Stammgruppen nach gegenseitiger Erklärung ein eigenes kleines Computernetzwerk zu planen und mithilfe von Postern dieses den anderen Gruppen zu präsentieren. Durch die konkrete Identifikation mit eigenen Netzwerkkomponenten zu Hause konnten auch zeitlich relativ anspruchsvolle Arbeitsaufträge gegeben werden, da die Schülerinnen und Schüler großes Interesse daran zeigten, „*wie das alles funktioniert*“.

3.3 Untersuchung von Eigenschaften eines Computernetzes

- Erkunden gängiger Netzwerkeigenschaften durch eigenständige Erforschung von Netzwerken mit Standardprogrammen
- Geografisches Zuordnen von IP-Adressen bzw. Nachverfolgung der Route von Nachrichtenpaketen
- Theoretische und praktische Möglichkeiten zur Anonymisierung in Computernetzen

In den folgenden beiden Unterrichtsstunden wird zunächst die Untersuchung von Netzwerkeigenschaften mit Standardbefehlen (*arp*, *ping*, *traceroute*, *nslookup* etc.) zur Nachvollziehbarkeit von Kommunikation durchgeführt, bevor theoretische und praktische Möglichkeiten (z.B. über das *TOR*-Netzwerk² zur Verschleierung auch von Verbindungsdaten) der Anonymisierung thematisiert bzw. ausprobiert wurden. Hierbei kann auch der Aufbau des Internets durch sogenannte Autonome Systeme thematisiert werden, siehe dazu [He15].

Die Thematik der Verschlüsselung wird dabei über die Visualisierung der Inhalte unverschlüsselten Netzverkehrs mit Werkzeugen wie *Wireshark*³ angesprochen (vergl. [HNR06, S.127], siehe Abb. 1. Die (komplexe) Technik hinter kryptographischen Verfahren kann in einer weiteren Unterrichtseinheit behandelt werden, siehe etwa *RSA&Co* von Witten und Schulz [WS10] oder im Kontext von Emails in [GHW12]. Weitere Anregungen zu Datenschutz und Privatsphäre finden sich in [Be15].

3.4 Auswirkungen der Vorratsdatenspeicherung

- Argumente für und gegen die Nachverfolgbarkeit von IP-Adressen im Internet
- Positionen verschiedener gesellschaftlicher Gruppen zur Vorratsdatenspeicherung
- Vorbereitung und Durchführung einer Podiumsdiskussion

² <https://www.torproject.org/>

³ <https://www.wireshark.org/>



Abb. 1: Mitlesen von vermeintlich geschützten Inhalten mit *Wireshark*.

Die in der abschließenden Doppelstunde vorbereitete und durchgeführte Podiumsdiskussion über die Vorratsdatenspeicherung sollte dabei einen groben Querschnitt durch die Gesellschaft repräsentieren, als Gruppen bieten sich hierbei etwa die Sicht der Provider, der Polizei, von Parteien und die von Datenschützern an. Wichtig dabei ist die konkrete Auseinandersetzung in Gruppen mit zugehörigem Quellenmaterial, z.B. Zeitungsartikel, Interviews, Pressemeldungen, Webseiten etc. Nur daraus können die Schülerinnen und Schüler sich in die gewählte Position hineinfinden, die Argumentationsketten nachvollziehen und technisch beurteilen, siehe Abb. 2 für einige Beispiele⁴

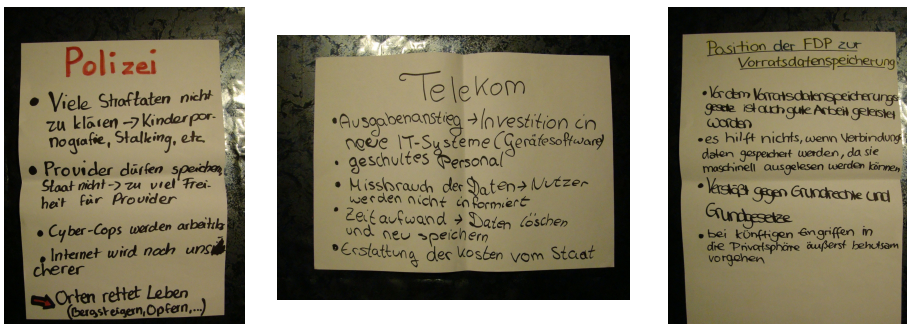


Abb. 2: Beispielposter für verschiedene gesellschaftliche Positionen zur Vorratsdatenspeicherung.

Bewährt hat sich dabei die Moderation der Podiumsdiskussion durch eine(n) Schülerin bzw. Schüler, da die Lehrkraft sich damit im (beobachtenden) Abseits aufhalten kann und es eher um die Sache an sich geht, wodurch nach kurzer Zeit auch sonst eher ruhigere Schülerinnen und Schüler in den Diskussionsfluss kamen.

Bemerkenswert war dabei die relative Länge der Podiumsdiskussionen (ca. 30 Minuten)

⁴ Ein alternativer Einstieg bietet sich über mögliche Auswertungsszenarien der Vorratsdaten an [OG14, OS11].

und die inhaltliche Tiefe, welche wahrscheinlich nur durch die direkte Identifikation im Rollenspiel, das geweckte persönliche Interesse der Schülerinnen und Schüler im Publikum und die vorherige ausführliche Quellenarbeit möglich war, wobei als direkte Grundlage auf die technischen Aspekte von Computernetzen zurückgegriffen werden konnte.

Auch ergab sich im Anschluss nicht ein einseitiger Ausschlag dafür oder dagegen, die einzelnen Schülermeinungen blieben differenziert, wenn auch eher gegen die Vorratsdatenspeicherung tendierend.

4 Fazit

Obwohl oder gerade weil die ersten Stunden stark theoretisch und technisch fundiert waren, zeigten die Schülerinnen und Schüler starkes Interesse und Begeisterung an der Thematik. Erst das technische Grundverständnis von Kommunikation in Computernetzen erlaubte ihnen eine angemessene Beurteilung der in Bezug stehenden gesellschaftlichen Aspekte der Informatik.

Nach Aussage mehrerer Schülerinnen und Schüler ergab sich aus der Unterrichtssequenz heraus zumindest ein kurzer Austausch mit der Familie, wobei gerade die Weitergabe technischen Wissens von Interesse war. Da das Hauptlernziel natürlich nicht eine Multiplikatoren- und Kompetenzbildung ist, sondern der Wissens- und Kompetenzzugewinn der Schülerinnen und Schüler, positiv überprüft durch eine Klassenarbeit, ist diese Weitergabe ein interessanter zusätzlicher Effekt, den wir in der Zukunft auch weiter untersuchen möchten.

Als eine weitere Auswirkung ergab sich auch, dass die Schülerinnen und Schüler nach der Sequenz viele Arten von eingerichteten Internetseitensperren (nicht nur) in der Schule umgehen konnten – und dieses Wissen als Multiplikatoren auch gerne aktiv unter Mitschülern verbreiteten – ein (un[?])gewollter Aspekt der Unterrichtssequenz.

Es wäre dabei sicherlich interessant, diese Multiplikatorenwirkung bewusst als Unterrichtsmethode zu verwenden, etwa durch die Einbeziehung der Schülerinnen und Schüler: Welches Wissen und welche Kompetenzen müsstet ihr anderen (Erwachsenen) vermitteln, sodass diese fachlich fundierte Entscheidungen treffen können?

Dabei ist die Vorratsdatenspeicherung nur ein (immer noch) aktuelles Anwendungsbeispiel für den Unterricht. Mögliche Alternativen im Bereich Computernetze wären etwa auch Netzneutralität oder Internetsperren.

Abschließend möchten wir noch einmal anmerken, dass (nicht nur) unserer Meinung nach *jede* Unterrichtssequenz, die Informatik und Gesellschaft behandelt, auch zwingend die technischen Aspekte behandeln muss. Denn zum Verstehen und Gestalten der digitalen Welt sind die fachlichen Hintergründe immer notwendig, sonst wandelt sich das Potenzial vom aktiven Mitwirken zum passiven Erfahren der Auswirkungen, ein gesellschaftlicher Aspekt, den nur das Schulfach Informatik adäquat behandeln kann.

Danksagung

Abschließend möchte der Autor herzlichst Ylva Brandt, Eckart Modrow und Kerstin Strecker für viele anregende Diskussionen und Hinweise danken, ebenso Ira Diethelm für das Konzept der Abendbrotgespräche und Vorabversionen ihrer Artikel sowie den anonymen Reviewern für ihre hilfreichen Anmerkungen, u.a. für die Idee der Multiplikatorenwirkung als Unterrichtsmethode, als auch Nina Schneider für Hinweise zur finalen Formatierung.

Literatur

- [Be15] Berendt, Bettina; Dettmar, Gebhard; Esslinger, Bernhard; Gramm, Andreas; Grillenberger, Andreas; Hug, Alexander; Witten, Helmut: Datenschutz im 21. Jahrhundert - Ist Schutz der Privatsphäre (noch) möglich? In: INFOS. Jgg. P-249 in LNI. GI, S. 33–42, 2015.
- [Bu04] Bundeszentrale für politische Bildung: , Makromethode Podiumsdiskussion (Forschen mit GrafStat). <http://www.bpb.de/lernen/formate/methoden/46894/podiumsdiskussion>, 2004. Zuletzt abgerufen am 10. Februar 2017.
- [DD11] Diethelm, Ira; Dörge, Christina: Zur Diskussion von Kontexten und Phänomenen in der Informatikdidaktik. In: INFOS. Jgg. P-189 in LNI. GI, S. 67–76, 2011.
- [Di07] Diethelm, Ira: Strictly models and objects first: Unterrichtskonzept und -methodik für objektorientierte Modellierung im Informatikunterricht. Dissertation, University of Kassel, Germany, 2007.
- [DKW11] Diethelm, Ira; Koubek, Jochen; Witten, Helmut: IniK – Informatik im Kontext: Entwicklungen, Merkmale und Perspektiven. LOG IN, 32/33(169/170):97–105, 2011.
- [Fo07] Fothe, Michael: Algorithmen in spielerischer Form. In (Stechert, Peer, Hrsg.): Praxisband der 12. Fachtagung Informatik und Schule INFOS. 2007.
- [Fo10] Foerster, Klaus-Tycho: Didaktische Reduktion des Themenkomplexes Computernetze unter besonderer Berücksichtigung der technischen Aspekte - ein Unterrichtsversuch im Jahrgang 9. Schriftliche Arbeit im Rahmen der zweiten Staatsprüfung, Studienseminar Göttingen für das Lehramt an Gymnasien, 2010.
- [GHW12] Gramm, Andreas; Hornung, Malte; Witten, Helmut: Email for you (only?): design and implementation of a context-based learning process on internetworking and cryptography. In: WiPSCE. ACM, S. 116–124, 2012.
- [Gi] Gierhardt, Horst: , CPU-Simulation als Rollenspiel (aus: Mikroprozessor-Simulation mit dem DC (Didaktischer Computer)). <http://www.gierhardt.de/informatik/dc/Rollenspiel.pdf>. Zuletzt abgerufen am 10. Februar 2017.
- [He15] Hennecke, Martin: Modellvorstellungen zum Aufbau des Internets. In: INFOS. Jgg. P-249 in LNI. GI, S. 155–164, 2015.
- [HNR06] Hartmann, Werner; Näf, Michael; Reichert, Raimond: Informatikunterricht planen und durchführen. Springer, 2006.
- [Hu07] Hubwieser, Peter: Didaktik der Informatik: Grundlagen, Konzepte, Beispiele (3.Auflage). Springer, 2007.

- [Ko] Koubek, Jochen: , Informatik im Kontext (IniK). <http://www.informatik-im-kontext.de>. Zuletzt abgerufen am 10. Februar 2017.
- [Me12] Merkert, Klaus: , Kommunikation in Rechnernetzen. <http://www.hsg-kl.de/faecher/inf/netze/lehrplan/index.php>, Dezember 2012. Zuletzt abgerufen am 10. Februar 2017.
- [Mo03] Modrow, Eckart: Fundamentale Ideen der theoretischen Informatik. In: INFOS. Jgg. 32 in LNI. GI, S. 189–200, 2003.
- [OG14] OpenDataCity; Glättli, Balthasar: , Vorratsdaten-speicherung in der Schweiz: Das überwachte Leben von Nationalrat Balthasar Glättli. <https://apps.opendatacity.de/vds/>, April 2014. Zuletzt abgerufen am 10. Februar 2017.
- [OS11] Online, Zeit; Spitz, Malte: , Vorratsdatenspeicherung: Verräterisches Handy. <http://www.zeit.de/datenschutz/malte-spitz-vorratsdaten>, März 2011. Zuletzt abgerufen am 10. Februar 2017.
- [SS04] Schubert, Sigrid E.; Schwill, Andreas: Didaktik der Informatik. Spektrum Akadem. Verl., 2004.
- [St11] Strecker, Kerstin: Einstieg in die Algorithmik. Lehrbildungszentrum Informatik Göttingen, 2011.
- [WS10] Witten, Helmut; Schulz, Ralph-Hardo: RSA&Co in der Schule - Moderne Kryptologie, alte Mathematik, raffinierte Protokolle. Neue Folge (Teil 4). LOG IN, 30(163/164):97–103, 2010.

Hamburg 2050, Land unter? Eine „Informatik im Kontext“-Reihe zu Klimamodellierung mit Geoinformationssystemen

Jonathan Otto¹, Michel Welke², Andreas Gramm³, Leonard Diekmann⁴,
Maria Knobelsdorf⁵

Abstract: Dieser Beitrag stellt eine interdisziplinär ausgerichtete „Informatik im Kontext“-Unterrichtsreihe für das Wahlpflichtfach Informatik in der Sekundarstufe I vor sowie erste Erfahrungen einer Erprobung der Reihe in Klasse 10 einer Hamburger Stadtteilschule. An die Kontextorientierung im Hamburger Bildungsplan für das Wahlpflichtfach Informatik anknüpfend, thematisiert die Reihe Grundkonzepte informatischer Bildung aus den Kompetenzbereichen „Informationen und Daten“, „Informatiksysteme“ und „Informatik, Mensch und Gesellschaft“ im Kontext von Klimawandel und Klimamodellierung mit dem Geoinformationssystem QGIS. Fächerübergreifende Bezüge zum Schulfach Geografie werden durch das Thema Klimawandel, das digitale Werkzeug QGIS und die damit einhergehenden geografischen Fragestellungen hergestellt.

Keywords: Informatik im Kontext, IniK, Bildungsstandards, Wahlpflichtfach Informatik, Hamburg, Geographische Informationssysteme, GIS.

1 Einleitung

„Informatik ist per se fachübergreifend und fächerverbindend, deshalb ist Interdisziplinarität ein Grundsatz der Unterrichtsgestaltung. Das bedeutet, dass informatische Kompetenzen im Grunde nur in einem Unterricht erworben werden können, der von vornherein interdisziplinär angelegt ist.“ ([GI08], S. 10). Angesichts der im Rahmen der Digitalisierung immer stärker zunehmenden Durchdringung von Branchen, Disziplinen, privaten und öffentlichen Bereichen mit informatischen Methoden, Anwendungen und Systemen ist der interdisziplinäre Aspekt informatischer Bildung aktueller denn je. Doch was bedeutet es konkret, wenn Informatikunterricht über die Grenzen der eigenen Disziplin hinausgehen soll? Wie kaum ein anderer fachdidaktischer Ansatz greift „Informatik im Kontext“ (IniK) [KSSW09] die Forderung nach Interdisziplinarität durch

¹ Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg,
Jonathan.Otto@informatik.uni-hamburg.de

² Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg,
michel.welke@studium.uni-hamburg.de

³ Gymnasium Tiergarten, Altonaerstr. 26, 10555 Berlin, gramm@gymnasium-tiergarten.de

⁴ Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg,
diekmann@informatik.uni-hamburg.de

⁵ Universität Hamburg, Fachbereich Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg,
knobelsdorf@informatik.uni-hamburg.de

die Orientierung an lebensweltlichen Kontexten und den Bildungsstandards zweifach auf. Mit der Orientierung an Kontexten wird im IniK-Ansatz das Ziel verfolgt, für die Schülerinnen und Schüler bedeutsame und nachvollziehbare Zusammenhänge zwischen ihrer Lebenswelt und den Fachinhalten des Schulfachs herzustellen. Hierzu sollen lebensweltliche Fragestellungen, Phänomene und Probleme der Alltagswelt betrachtet werden, die verschiedene fachliche Dimensionen haben und so einen Bezug zu mehreren Fachdisziplinen ermöglichen. Weil nicht nur informatische, sondern auch z. B. politische, ökonomische, ethische oder rechtliche Fragestellungen eines Kontexts thematisiert werden, kann so ein interdisziplinärer Unterricht gestaltet werden. In den letzten Jahren sind verschiedene, vor allem im Berliner Raum entwickelte, IniK-Unterrichtsreihen vorgestellt worden, die beispielhaft erfolgreich das Prinzip des kontextorientierten Informatikunterrichts umgesetzt haben. Das Prinzip der Interdisziplinarität ist dabei jedoch nicht in dem Maße berücksichtigt worden, wie die Kontextkriterien von IniK dies jeweils vorsehen. Diese Lücke wollen wir schließen und im folgenden Artikel eine interdisziplinäre IniK-Unterrichtsreihe zu Klimawandel und -modellierung mit Geoinformationssystemen (GIS) vorstellen, die Bezüge zum Geografieunterricht in der Sekundarstufe I (Sek. I) herstellt. Im nächsten Abschnitt stellen wir kurz den fachdidaktischen Ansatz IniK vor und leiten dann her, inwiefern das Thema Klimawandel und -modellierung mit GIS als möglicher Kontext für eine IniK-Reihe umsetzbar ist. In Abschnitt 3 beschreiben wir die konkrete Unterrichtsplanung. Abschnitt 4 zieht ein erstes Fazit aus der konkreten Erprobung der Unterrichtsreihe an einer Hamburger Stadtteilschule.

2 Informatik im Kontext der Geografie

Als zentrale Merkmale eines IniK-Unterrichts werden die folgenden drei Prinzipien genannt: Orientierung an lebensweltlichen Kontexten, Orientierung an Bildungsstandards wie den Empfehlungen der Gesellschaft für Informatik (GI) (vgl. [GI08]) und der Einsatz einer Vielfalt an lernerzentrierten Methoden. Weiterhin werden fünf Kriterien für die Auswahl geeigneter Kontexte definiert (im Folgenden als „Kontextkriterien“ bezeichnet). Was bedeutet die Orientierung an Kontexten genau? Was bedeutet es für informatische Bildung und den Informatikunterricht, wenn ein Bezug zur Lebenswelt der Schülerinnen und Schüler gezogen wird und der Unterricht dabei interdisziplinäre Züge aufweisen soll? Koubek et al. (2009) definieren einen Kontext als einen spezifischen Handlungsrahmen einer konkreten, durch das Individuum erlebbaren Situation: „Jede Situation hat ihre technischen, sozialen, ethischen, rechtlichen, ökonomischen, ästhetischen etc. Aspekte, die zusammen den Kontext ausmachen.“ ([KSSW09], S. 4) Konkret auf den Unterricht bezogen stellen Kontexte eine „Menge von lebensweltlichen Themen bzw. Fragestellungen [dar], die von den Schülerinnen und Schülern als zusammenhängend geordnet werden und die dadurch sinnstiftend auf deren Handlungen wirken“ (ebd., S. 5). Ein solcher Kontext sollte laut Diethelm et al. ([DKW11], S. 102) informatisch relevant sein und sich auf möglichst viele Kompetenzen der Bildungsstandards beziehen (Kriterium der Tiefe). Er sollte aber

auch Bezüge zu anderen Disziplinen oder Bereichen aufweisen (Kriterium der Mehrdimensionalität) und gesellschaftlich relevant (Kriterium der Breite), über einen längeren Zeitraum stabil (Kriterium Stabilität) sowie durch Bezüge zur Lebenswelt der Schülerinnen und Schüler potenziell erlebbar (Kriterium der Lebenswelt) sein. Für die konkrete Unterrichtsstruktur wurde vorgeschlagen, dass sich eine IniK-Unterrichtseinheit an den vier Phasen kontextorientieren Unterrichts orientiert: Begegnungsphase, Neugier- und Planungsphase, Erarbeitungsphase und Vernetzungsphase [KSSW09]. Wir wollen im Folgenden das geographische Thema Klimawandel und sein geoinformatisches Pendant Klimamodellierung mit GIS als einen möglichen Kontext für eine IniK-Unterrichtsreihe vorstellen und diskutieren.

2.1 Klimawandel und Klimamodellierung mit GIS als lebensweltlicher Kontext

Die Geowissenschaften durchleben seit Jahrzehnten eine starke digitale Transformation, die unter anderem Geoinformationssysteme (GIS) hervorgebracht hat. GIS sind ein leistungsstarkes und vielfältiges digitales Werkzeug, um geografische Fragestellungen mit informatischen Methoden zu erforschen. Auch im Schulbereich sind GIS als ein zentrales Lehr-Lern-Werkzeug für den Geografieunterricht anerkannt und in die Geografie-Bildungsstandards für die Sek. I integriert worden [MSV11]. Klima, als ein zentrales Thema des Schulfaches Geografie in der Sek. I, beinhaltet ein Verständnis des Klimasystems an sich und analysiert dessen Interaktion mit anderen Ökosystemen auf der Erde. Der Schwerpunkt liegt dabei auf der Betrachtung der verschiedenen Klimaelemente (Temperatur, Niederschlag, Wind, Sonnenschein usw.) und wie deren Veränderungen das System Erde beeinflussen. Forschung in diesem Bereich ist von großer Bedeutung, um insbesondere den menschlichen Einfluss auf das Klimasystem zu verstehen. Hierfür werden computerbasierte Klimamodelle entwickelt und eingesetzt, um Veränderungen zu analysieren und Vorhersagen für die Zukunft treffen zu können. Für die Erfassung, Analyse und Visualisierung von Klimadaten stellen GIS ein wichtiges und vielgenutztes Werkzeug dar.

GIS bestehen aus einer räumlichen Datenbank und einem grafischen Interface zur Erfassung, Verwaltung, Analyse und Visualisierung von Daten (vgl. Abbildung 1), die einen räumlichen und zeitlichen Bezug haben. Raum- und zeitbezogene Datenquellen definieren geografische Informationen, die sich auf Objekte und Prozesse geografischer Phänomene beziehen, die auf der Erde beobachtbar sind. Geografische Phänomene können hinsichtlich ihrer Ursachen und Wirkungen variieren: So ist Niederschlag zum Beispiel ein Wetterphänomen, während Migration ein geopolitisches und sozio-kulturelles Phänomen darstellt. Die Datenvisualisierung mit einem GIS stellt eine mehrschichtige Repräsentation räumlich-zeitlicher Prozesse mit unterschiedlichen Dimensionen und Auflösungen dar, die weit über die Möglichkeiten einer Papierkarte hinausgehen. Die Darstellung der Daten wird durch raster- und vektorbasierte Grafiken umgesetzt, wobei unterschiedliche Datenformate sowie Möglichkeiten zur Datenanalyse bereitgestellt werden. Mithilfe von SQL (Structured Query Language) können Daten sowohl attributbezogen als auch räumlich selektiert und analysiert werden. Eine Reihe

unterschiedlicher Algorithmen sind in jedem GIS integriert und können durch eigene Programme (z. B. mit Python oder Java) ergänzt werden [SS05]. Die Software QGIS ist im Schulbereich ein vielgenutztes, Open-Source-basiertes GIS, welches alle gängigen Funktionen für die Arbeit mit GIS enthält.

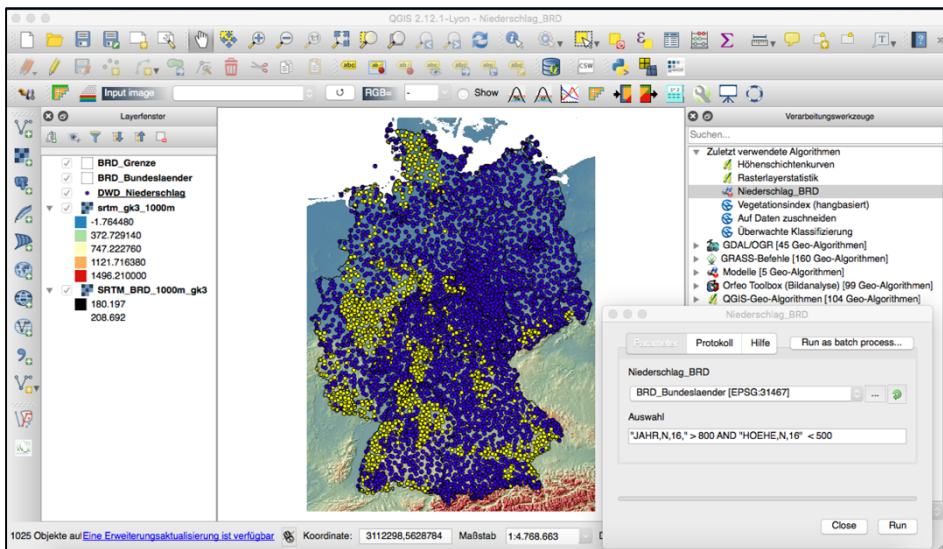


Abbildung 1: Screenshot der Software QGIS mit Niederschlagsdaten für Deutschland (Datenquelle: Deutscher Wetterdienst, DWD)

2.2 IniK-Kontextkriterien und gewählter Kontext

GIS an sich stellen noch keinen Lebensweltbezug für die Schülerinnen und Schüler dar und benötigen einen Anwendungsbereich, in dem sie sinnstiftend zur Erkenntnisgewinnung eingesetzt und genutzt werden können. GIS können in zahlreichen unterschiedlichen Kontexten zum Einsatz kommen, z. B. für die Raum- und Stadtplanung, das Management des Verkehrsaufkommens in Städten und Häfen, die Analyse von Naturkatastrophen oder die archäologische Kartografie von Ausgrabungsstätten. Der in dieser Unterrichtseinheit verwendete Kontext aus dem Fach Geografie ist das Thema Klima und Klimamodellierung am regionalen Beispiel von Hamburg. Der regionale Bezug stellt dabei das Klima und die Folgen des Klimawandels im Raum Hamburg in den Mittelpunkt. Aufgrund der gesellschaftlichen Relevanz von Klimaforschung in Zeiten stark wahrnehmbarer Klimaveränderungen und dem Einsatz von Informatiksystemen in diesem Bereich eignet sich Klimawandel und -modellierung unserer Meinung nach für einen kontextorientierten Informatikunterricht, der auch interdisziplinäre Bezüge zum Schulfach Geografie herstellt. Dieser Eindruck bestätigt sich, wenn man den Kontext bezüglich der fünf Kontextkriterien betrachtet.

Der Klimawandel und die Frage, welche Auswirkungen menschliches Handeln darauf hat, stellt nicht nur für die Schülerinnen und Schüler, sondern für die gesamte Weltbevölkerung einen lebensweltlichen Kontext dar, der durch Wetterphänomene sowie Diskussionen um Folgefaktoren wie Migrationsbewegungen oder Ressourcenkonflikte direkt und indirekt erlebbar ist. Der Kontext hat aufgrund der menschlichen Veränderungen des Klimasystems und der damit verbundenen Verantwortung im Hinblick auf Produktion und Nachhaltigkeit eine ethische und ökonomische Dimension sowie aufgrund der räumlichen Wirkung auf verschiedenen Skalen auch eine geografische Dimension. Die Klimamodellierung und -simulation mit informatischen Werkzeugen wiederum ermöglicht überhaupt erst eine fundierte wissenschaftliche Betrachtung des Themas, wodurch informatische Methoden, Konzepte und Prinzipien eine zentrale Rolle spielen. Damit ist das Kriterium der Mehrdimensionalität im Sinne des IniK-Ansatzes erfüllt und Bezüge zu einem fächerübergreifenden Unterricht sind gegeben. So kann eine IniK-Reihe mit Schwerpunkt auf Klimamodellierung mit GIS im Informatikunterricht starten und im Schulfach Geografie mithilfe von GIS weitere Fragestellungen bearbeiten. Darüber hinaus ist in Abhängigkeit von der Fragestellung die Integration in weitere Fächer möglich (z. B. Politische Weltkunde, Biologie, Physik), die eine nicht-informatische Dimension haben. Neben dem Kontextkriterium der Breite ist auch das Kriterium der Tiefe erfüllt: Wie das nächste Kapitel zeigen wird, ist für die Betrachtung und Entwicklung von Klimamodellen und -simulationen ein solides Hintergrundwissen der Informatik notwendig, welches Kompetenzen aus den Bereichen der Datenverwaltung, Algorithmik, Programmierung und Modellierung erfordert, womit Kompetenzen aus mehreren Kompetenzbereichen der GI-Bildungsstandards erworben werden. An dieser Stelle wird deutlich, dass ohne den thematischen Kontext GIS lediglich eine Datenbankanwendung ist, ohne spezifischen Bedeutungsrahmen für die Schülerinnen und Schüler der Sek. I. Erst durch den Handlungsrahmen, den das Thema Klimawandel und -modellierung vorgibt, entsteht ein mehrdimensionaler Kontext, der interdisziplinäre Fragestellungen für die Informatik und die anderen Schulfächer aufwirft.

3 Eine IniK-Unterrichtsreihe zu Klimawandel mit QGIS

In diesem Abschnitt skizzieren wir eine mögliche IniK-Unterrichtsreihe zur Erforschung des Klimawandels mit einem GIS. Wie für IniK-Reihen vorgeschlagen, orientiert sich die informatische Dimension der hier beschriebenen Unterrichtsreihe an den Empfehlungen der GI (Gesellschaft für Informatik e. V.) zu Bildungsstandards für die Sek. I. Konkret werden Kompetenzen aus den Inhaltsbereichen „Informationen und Daten“, „Informatiksysteme“ und „Informatik, Mensch und Gesellschaft“ für die Klassenstufe 8 bis 10 erworben ([GI08], S. 12ff.). Die Unterrichtsreihe fördert dabei besonders folgende Aspekte:

- **Informationen und Daten:** Informationen in unterschiedlicher Form darstellen, Daten im Kontext der repräsentierten Information interpretieren und

Vor- und Nachteile unterschiedlicher Informationsdarstellungen beurteilen können.

- **Informatiksysteme:** Das Betriebssystem zweckgerichtet benutzen, Dateiformate unterscheiden und sich selbstständig neue Anwendungen und Informatiksysteme erschließen können.
- **Informatik, Mensch und Gesellschaft:** Seriosität und Authentizität von Informationen aus dem Internet beurteilen, Auswirkungen der Automatisierung auf die Arbeitswelt bewerten und Beurteilen der Umsetzung automatisierter Vorgänge.

In der Planung der Unterrichtsreihe stellten sich drei Aspekte als eine Herausforderung dar: 1) die Einführung in den thematischen Kontext, 2) Einführung des Werkzeugs QGIS und 3) die Dekontextualisierung hin zu den Fachkonzepten der Bildungsstandards. Konkrete Fragen stellten sich insbesondere zur Gewichtung, wie z. B. Wie viel Kontext ist nötig, um das Werkzeug und die Fachkonzepte zu motivieren? Wie steht es um das Vorwissen der Schülerinnen und Schüler im Hinblick auf den Klimawandel? Wie intensiv muss das Werkzeug eingeführt werden, um die eigentlichen Leitfragen des Kontexts bearbeiten zu können? Hier wurde relativ schnell deutlich, dass die Planung sich nur sehr grob an den IniK-Unterrichtsphasen orientieren kann. In den folgenden Unterabschnitten beschreiben wir die einzelnen Unterrichtsphasen der hier vorgestellten Unterrichtsreihe zu Klimamodellierung mit QGIS, wobei wir auf die genannten Fragen eingehen.

3.1 Klimawandel als thematischer Kontext

Zu Beginn der IniK-Unterrichtsreihe, wie wir sie hier im Weiteren skizzieren wollen, steht zunächst der Kontext Klimawandel und die geografische, ökonomische sowie ethische Dimension dieses Kontexts im Mittelpunkt. Dazu gehört auch die Relevanz des Klimawandels als globales und regionales Phänomen, welches wir im Folgenden konkret an der Stadt Hamburg und damit als eine mögliche Unterrichtseinheit vor allem aber nicht nur für Hamburger Schülerinnen und Schüler betrachten wollen. Die informatische Dimension ist in dieser Phase zunächst zweitrangig, da es zunächst darum gehen soll, den Bezug zur Lebenswelt der Schülerinnen und Schüler herzustellen und dabei vor allem die ökologischen, ökonomischen und sozialen Folgen des Klimawandels, am Beispiel von Hamburg, zu thematisieren. Die Länge dieser Phase ist abhängig vom Vorwissen der Schülerinnen und Schüler. Sofern diese im Fach Geografie die klimageografischen Grundlagen gelernt haben, kann direkt mit einem motivierenden Beispiel eingestiegen werden, das die entsprechende Leitfrage(n) der Unterrichtsreihe motiviert. Ein solches Beispiel kann Informationsmaterial mit für die Schülerinnen und Schüler regional relevanten Klimaprognosen sein. Am Beispiel Hamburgs wären dies z. B. Überflutungsszenarien, wie sie bereits durch Sturmfluten in Hafennähe bekannt sind. Hierzu schlagen wir konkret vor, mit einem Video zum Thema Klimawandel und einer anschließenden Diskussion im Klassenplenum zu starten. Daraufhin können die

Schülerinnen und Schüler, z. B. im Rahmen einer Gruppenarbeit, ein aktuelles Zeitungsinterview eines Klimaforschers auswerten, das sich mit einer Klimaprognose auseinandersetzt. Wir haben hierzu z. B. Material gefunden, das eine komplette Überflutung von Hamburg im Jahr 2050 prognostiziert. Untermalt von Überflutungssimulationen sowie echten Fotos vom überfluteten Hamburger Fischmarkt soll ein solches regionales Beispiel und die in diesem Fall sehr extreme Prognose, die potenziellen Auswirkungen des Klimawandels auf das Lebensumfeld der Schülerinnen und Schüler verdeutlichen und für die weitere Auseinandersetzung sinnstiftend wirken. Die weitere(n) Leitfrage(n) der Unterrichtsreihe können eine solche Prognose in Frage stellen und für die Schülerinnen und Schüler den Arbeitsauftrag enthalten, zu untersuchen, inwiefern die Prognose tatsächlich realistisch wäre.

3.2 Klimamodellierung und Kennenlernen von QGIS

An die Begegnung mit dem thematischen Kontext Klimawandel folgt im Anschluss die erste Auseinandersetzung mit Klimamodellierung, die auch das Kennenlernen des Werkzeugs QGIS umfasst. In dieser Phase sollten die Schülerinnen und Schüler den grundlegenden Aufbau des Werkzeugs QGIS sowie die wichtigsten Funktionen und das Layerprinzip (Darstellung der räumlichen Daten in übereinanderliegenden Ebenen) kennenlernen, um im weiteren Verlauf das Werkzeug für Fragestellungen zur Klimamodellierung einsetzen zu können. Neben der reinen Nutzung des Werkzeugs QGIS wird in dieser Phase auch schon eine erste Dekontextualisierung in Bezug auf die dem Werkzeug zugrundeliegenden informatischen Inhalte vorgenommen. Dabei sollen die Schülerinnen und Schüler das Begriffspaar „Informationen und Daten“ kennenlernen und voneinander abgrenzen sowie mithilfe einer grafischen Darstellung die Informationsverarbeitung am Beispiel von Klimadaten visualisieren können. Im Rahmen dieser Phase stehen vor allem die technische (Nutzung des Werkzeugs) sowie die informatische Dimension (Daten als Informationsträger) im Mittelpunkt. Die geografische Dimension als Kontext ist immer noch präsent, spielt in dieser Phase jedoch eine untergeordnete Rolle. In der fachlichen Tiefe liegt der Schwerpunkt dieser Phase auf den Kompetenzbereichen „Informationen und Daten“ und „Informatiksysteme“. Während bei dem Kompetenzbereich „Informationen und Daten“ vor allem die Darstellung der Daten sowie die Interpretation der zugrundeliegenden Repräsentation im Vordergrund steht, liegt der Fokus bei den „Informatiksystemen“ auf der zweckgerichteten Nutzung und der Unterscheidung unterschiedlicher Datentypen.

Um diese Phase entsprechend zu motivieren, schlagen wir vor, die Schülerinnen und Schüler in einer Partnerarbeit eine eigene Visualisierung ihrer Vorstellungen zum Thema „Hamburg im Jahr 2050“ anfertigen zu lassen. Das Ziel hierbei ist, dass die Schülerinnen und Schüler mit ihrem Ergebnis unzufrieden sind, da die ihnen bekannten, gängigen Visualisierungsmöglichkeiten starke Grenzen setzen, um Klimamodellierung und konkrete Prognosen adäquat darzustellen. In einem Gespräch im Klassenplenum können die Potenziale und Grenzen der bisherigen Visualisierungsmöglichkeiten an der Tafel gesammelt und diskutiert werden. Im Anschluss daran findet die Einführung von QGIS

statt. Hier schlagen wir vor, dass die Schülerinnen und Schüler das Werkzeug selbst mithilfe eines bereitgestellten Tutorials kennenlernen. Wir haben hierzu ein Tutorial erstellt, welches aus mehreren Arbeitsblättern besteht und sequenziell abgearbeitet wird, wobei die Grundfunktionen des Werkzeugs sowie das für die Arbeit mit QGIS wichtige Layerprinzip vorgestellt werden.

Für die konkrete Auseinandersetzung mit QGIS können sich die Schülerinnen und Schüler mit Klima- und Wetterdaten auseinandersetzen, wie sie z. B. durch den Deutschen Wetterdienst (www.dwd.de, <ftp://ftp-cdc.dwd.de/pub/CDC/>) oder auf dem Hamburger Bildungsserver bereitgestellt werden (<http://bildungsserver.hamburg.de/daten-zum-klimawandel/>). Tatsächliche Aufgaben wären z. B. das Modellieren einer digitalen Karte in QGIS auf Grundlage dieser Daten. Hierbei müssen Objekte festgelegt und ihre Attribute zugewiesen werden (in diesem Fall Werte verschiedener Klimaelemente, z. B. Temperatur, Niederschlag usw.). Im Vordergrund stehen dabei auch die verwendeten Datentypen (String, Integer, Dezimalzahlen) für die einzelnen Klimaelemente und deren Vor- und Nachteile. Ebenso relevant sind die grafischen Repräsentationsformen der Vektor- und Rastergrafik, die die Visualisierung der Modellierung letztendlich ermöglichen. Die entsprechenden informatischen Konzepte werden dabei konkret thematisiert und so im Weiteren auch dekontextualisiert betrachtet.

3.3 Erarbeitungsphase und Abschluss

In einer anschließenden Projektphase werden die Leitfrage(n) aus der Begegnungsphase der Unterrichtsreihe im Rahmen einer Projektarbeit wieder aufgegriffen und die in der Phase vorher erarbeiteten Kenntnisse und Fähigkeiten angewendet und vertieft. Das zu bearbeitende Projekt sollte sich mit der Visualisierung verschiedener Klimaelemente im gewählten, regionalen Raum mithilfe von QGIS auseinandersetzen, mit dem Ziel, prognostizierte Auswirkungen des Klimawandels zu analysieren und darzustellen. Hierzu müssten entsprechende Datensätze und die geeignete Repräsentationsform ausgewählt und in QGIS entsprechend der Leitfrage(n) modelliert werden. Für unser regionales Beispiel Hamburg schlagen wir vor, dass die Schülerinnen und Schüler als Projektaufgabe eine eigene Karte zum Thema Klimawandel in Hamburg erstellen, mit der sie die Prognose des Klimaforschers aus der Begegnungsphase entweder belegen oder widerlegen. Dazu können sie in Partnerarbeit ein oder mehrere Klimaelemente (Temperatur, Niederschlag usw.) und die für die Fragestellung am besten geeignete grafische Repräsentationsform auswählen. Die von den Schülerinnen und Schülern bearbeiteten Projekte können im Anschluss an diese Phase in der Klasse präsentiert und besprochen werden. Der Schwerpunkt der Diskussion sollte dabei auf der Wahl der Repräsentationsform, der Qualität der Daten und der Visualisierung liegen und welche Verbesserungsmöglichkeiten es hinsichtlich dieser geben würde.

4 Erfahrungen und erstes Fazit

Die in Abschnitt 3 skizzierte IniK-Unterrichtsreihe wurde von uns in einer 10. Klasse einer Hamburger Stadtteilschule im Rahmen des Wahlpflichtfachs Informatik im Umfang von 14 Unterrichtsstunden praktisch erprobt. Für diese Erprobung wurde die Reihe zusätzlich zu den Bildungsstandards auch an den Hamburger Bildungsplan angepasst. Hierzu wurden die Module „Prozesse analysieren und modellieren“ und „Daten und Prozesse“ ausgewählt. Ein Fokus lag dabei auf Repräsentation von Informationen, Algorithmen und prozedurale Programmierung, Exploration des gewählten Anwendungskontextes, Analyse von Einsatzmöglichkeiten eines Informatiksystems im gewählten Anwendungskontext und Daten strukturieren sowie Variablen und Parameter verwenden ([FFH14], S. 22ff). Die Klasse bestand aus insgesamt 19 Schülerinnen und Schülern, wobei die Voraussetzungen und die Leistungsstärke sehr heterogen verteilt waren. So hatten z. B. einige Schülerinnen und Schüler schon im vorhergehenden Schuljahr Informatikunterricht, in dem sie sich mit Konzepten wie dem sequenziellen Durcharbeiten eines Problems zur Problemlösung auseinandergesetzt hatten. Auch Grundkenntnisse zum Thema Klimawandel waren unterschiedlich stark ausgeprägt. Zusätzlich strebten die Schülerinnen und Schüler des Kurses unterschiedliche Schulabschlüsse an (sowohl das Abitur als auch den ersten bzw. den mittleren allgemeinbildenden Schulabschluss), sodass auf eine entsprechende Binnendifferenzierung besonders intensiv eingegangen werden musste. Umgesetzt wurde dies durch eine vom Lehrer gesteuerte Partnerzusammenstellung der Projektgruppen während der ersten zwei Unterrichtsphasen auf Grundlage des Leistungsstands. In der dritten Phase konnten die Schülerinnen und Schüler ihre Projektpartner selbst wählen und erhielten eine gestufte Hilfestellung von Seiten der Lehrkraft. Ebenso wurde die Projektarbeit selbst durch die Lehrkraft stärker angeleitet. Die Lehrkraft gab dabei sowohl die Fragestellung als auch die zur Verfügung stehenden Daten vor.

Die Erprobung der Unterrichtsreihe wurde von Mitte Januar bis Anfang März 2017 durchgeführt. Der Einstieg über den Kontext Klimawandel und der regionale Bezug am Beispiel von Hamburg verlief erfolgreich und hat die Schülerinnen und Schüler von Beginn an stark motiviert. Die eigene Visualisierung im Rahmen der Begegnungsphase lieferte das gewünschte Ergebnis einer übereinstimmenden Unzufriedenheit mit den vorhandenen Möglichkeiten der Darstellung einer eigenen Prognose. Dabei wurden von den Schülerinnen und Schülern mehrere Vorschläge gemacht, wie man die Möglichkeiten der Visualisierung verbessern könnte. Die Vorschläge umfassten unter anderem eine kartografische Darstellung und Animationen des Klimawandels, wie sie durch QGIS umgesetzt werden können. Die eigenständige Einführung in die Arbeit mit QGIS mithilfe des Tutorials verlief erfolgreich und die Schülerinnen und Schüler waren bei der Bearbeitung der Aufgaben sehr motiviert und engagiert. Dabei konnten wir insbesondere beobachten, dass eher zurückhaltende Schülerinnen und Schüler deutlich motivierter und aktiver waren als in der vorhergehenden Unterrichtseinheit zum Thema Robotik. Die Doppelstunde zum Thema „Information und Daten“ lief wie geplant und

zeigte deutlich, dass die anfänglich wahrgenommene Motivation im Rahmen von Kontext und Werkzeug auch in dieser Phase stark zu erkennen war. So gaben viele der Schülerinnen und Schüler immer wieder Beispiele aus dem thematischen Kontextbereich, um die Begriffe Information und Daten zu erklären. Weiterhin konnten die Schülerinnen und Schüler begründete Entscheidungen treffen, welche Datentypen für die Repräsentation der jeweiligen Klimaelemente am besten geeignet sind. Ausgehend vom Kontext gelang es den Schülerinnen und Schülern problemlos viele weitere Beispiele aus anderen Bereichen zu nennen, für die sich eine Repräsentation durch Gleitkommazahlen, Integer oder auch Wahrheitswerte eignet. Die hier zu erkennende erste Vernetzung führte zu einer sehr regen Diskussion.

Unsere ersten Erfahrungen im Rahmen einer IniK-Unterrichtsreihe zum Thema Klimamodellierung mit GIS zeigen, dass das Thema und Werkzeug ein geeigneter Kontext sein kann, um Konzepte informatischer Bildung in einen Bezug zur Lebenswelt der Schülerinnen und Schüler zu setzen und sie zu motivieren, sich damit zu beschäftigen. Das Werkzeug QGIS scheint sich hier besonders gut zu eignen, um eine Auseinandersetzung mit informatischer Bildung im Kontext und am Schnittpunkt zum Schulfach Geografie zu motivieren.

Literaturverzeichnis

- [DKW11] Diethelm, I.; Koubek, J.; Witten, H.: IniK – Informatik im Kontext, Entwicklungen, Merkmale und Perspektiven. LOG IN, 169/170, S. 97-105, 2011.
- [GI08] Gesellschaft für Informatik e. V. (GI): Grundsätze und Standards für die Informatik in der Schule: Bildungsstandards Informatik für die Sekundarstufe I - Empfehlungen der Gesellschaft für Informatik e. V. Beilage zur LOG IN, 28, 150/151, 2008.
- [KMK16] Kultusministerkonferenz (KMK): Bildung in der digitalen Welt, Strategie der KMK, 2016.
- [KSSW09] Koubek, J.; Schulte, C.; Schulze, P.; Witten, H.: Informatik im Kontext (IniK) – Ein integratives Unterrichtskonzept für den Informatikunterricht. Im Tagungsband der 13. GI-Fachtagung Informatik und Schule. Köln Verlag, S. 268–279, 2009.
- [MSV11] Michel, U.; Siegmund, A.; Volz, D. Digitale Revolution im Klassenzimmer?! Potenziale digitaler Geomedien für einen zeitgemäßen Unterricht. In: Praxis Geographie, Braunschweig, 2011.
- [SS05] Strahler, A.; Strahler, A.: Introducing Physical Geography, Wiley, 2005.
- [FFH14] Freie und Hansestadt Hamburg: Bildungsplan Stadtteilschule Jahrgangsstufen 7 – 11 Informatik Wahlpflichtfach. 2014.

Musikprogrammierung mit Sonic Pi. Entwicklung und Untersuchung einer gendersensiblen Unterrichtseinheit zum Programmieren in der Sekundarstufe I.

Esther Alzate Romero¹, Leonore Dietrich²

Abstract: Die Musikprogrammiersprache Sonic Pi³, die an der University of Cambridge speziell für Kinder und Jugendliche entwickelt wurde, eröffnet einen kreativen, kontextbasierten Zugang zur Informatik. Ganz nach dem Motto „Technik zum Anfassen“ können durch den Einsatz der günstigen Hardwarelösung Raspberry Pi Berührungspunkte zu Informatiksystemen abgebaut werden. Mithilfe des Design-Based-Research-Ansatzes wird eine Unterrichtseinheit zum Thema Musikprogrammieren unter gendersensiblen Aspekten entwickelt und untersucht. Im vorliegenden Beitrag wird vor allem die erste Praxisphase, die daraus resultierenden Erkenntnisse und damit verbundenen Verbesserungs- bzw. Überarbeitungsschritte thematisiert. Hierbei stehen vor allem die Gestaltungsaspekte der Unterrichtseinheit im Vordergrund, ebenso die Frage, welche zentralen Inhalte und Konzepte der Informatik in diesem Rahmen vermittelt werden können. Zum Abschluss wird ein Ausblick auf das weitere Vorgehen gegeben.

Keywords: Informatikdidaktik, Genderforschung, Unterrichtsforschung, Musikinformatik, Live-Coding, Sonic Pi, Design-Based-Research, ProgrammierEinstieg, Informatik im Kontext.

1 Motivation und Forschungsvorhaben

Das ungleiche Geschlechterverhältnis in MINT-Berufen ist auch heute noch ein ernstzunehmendes Thema.

Betrachtet man die aktuellen Zahlen aus dem “MINT-Herbstreport 2016” des Instituts der deutschen Wirtschaft Köln, so gibt es trotz der hohen Zuwanderungszahlen immer noch eine “Arbeitskräftelücke” (212.000 unbesetzte Stellen) im MINT-Bereich [IWK16]. Dies stellt ein ernstzunehmendes Problem dar und es ist dringend vonnöten jungen Menschen diesen Arbeitsbereich näher zu bringen.

Aus Sicht der Gesellschaft ist die überwiegend männliche Gestaltung unserer zunehmend IT-gesteuerten Welt und Gesellschaft problematisch. Informatiksysteme werden auch künftig immer weiter in die Gestaltung aller Bereiche unserer Welt eingreifen – je weiter dieser Einfluss geht, desto wichtiger ist eine breite Repräsentanz aller Gesellschaftsgruppen in Berufen, die diese Gestaltung beeinflussen [Sc12]. Daher

¹ Heidelberg School of Education, Voßstraße 2, Geb. 4330, 69115 Heidelberg, romero@heiedu.ph-heidelberg.de

² Universität Heidelberg, Didaktik der Informatik, INF 205, 69120 Heidelberg, leonore.dietrich@uni-heidelberg.de

³ <http://sonic-pi.net> [25.01.2017]

gilt es Maßnahmen zu entwickeln, um Mädchen für informatische Fragestellungen zu begeistern.

Musik spielt im Leben von Jugendlichen eine große Rolle – Informatiksysteme werden von diesen aber in aller Regel nur zum Konsum genutzt. Bei Mädchen ist das Rezipieren und Ausüben von Musik tendenziell stärker vertreten als bei Jungen [Me16]. Dieses Interesse soll im Forschungsvorhaben genutzt werden, um innerhalb des Musikunterrichts in das Thema Programmierung einzuführen und damit im Kontext des Fachs Musik das Interesse an Informatik zu wecken. Durch die speziell für Kinder und Jugendliche ab etwa 10 Jahren entwickelte Entwicklungsumgebung Sonic Pi sollen zentrale Konzepte und Prozesse der Informatik vermittelt werden. Zendler und Spannagel [ZS06] arbeiten die zentralen Konzepte durch eine empirische Grundlegung heraus, um sich dem „Kern“ des Unterrichtsfaches Informatik anzunähern. Ebenso erheben sie im Rahmen einer Befragung unter Expert_innen die zentralen Prozesse im Informatikunterricht [Ze07]. Die Verknüpfung der zentralen Konzepte und Prozesse hilft bei der Gestaltung der Unterrichtseinheit. Als für die Unterrichtseinheit relevante zentrale Konzepte dieser Studie wurden die Konzepte Algorithmus, Problem, Sprache, Kommunikation, Computer und (Informatik-)System identifiziert. Die hier abgedeckten Prozesse aus der Studie sind das Problemlösen, Kreieren und Erfinden sowie Analysieren. Die GI Standards [GI08] für Informatik in der Sekundarstufe I weisen „Information und Daten“, „Algorithmen“ und „Sprachen und Automaten“ als für dieses Projekt relevante Inhaltsbereiche sowie „Modellieren und Implementieren“, „Strukturieren und Vernetzen“ und „Darstellen und Interpretieren“ als relevante Prozessbereiche aus.

Im Fokus des ersten Teils des Forschungsvorhabens steht zunächst die Entwicklung, Weiterentwicklung und Untersuchung der Unterrichtseinheit. Zudem wird untersucht, wie eine Unterrichtseinheit zum Thema Musikinformatik gestaltet werden muss, damit Mädchen und Jungen gleichermaßen für das Fach Informatik begeistert werden. Im zweiten Teil des Forschungsvorhabens wird die gendersensible Unterrichtsgestaltung im Vordergrund stehen. Grundlage für die Entwicklung der Unterrichtseinheit ist zum einen eine ausführliche Analyse der vorhandenen Literatur und Theoriekonzepte. Zum anderen wird den oben genannten Fragestellungen im Rahmen qualitativer Untersuchungen nachgegangen. Neben Interviews mit den beteiligten Schüler_innen und Lehrer_innen werden Expert_inneninterviews geführt.

Es gibt kaum Forschungsarbeiten zu Unterrichtseinheiten mit Bezug zur Musikinformatik. Folglich erfordert diese Forschung ein innovatives und damit ein auf die Weiterentwicklung und Erprobung fokussiertes Vorgehen. Daher wurde Design-Based-Research als Forschungsmethode gewählt. Das Konzept wird in mehreren Iterationen erprobt und nach jeder Erprobungsphase angepasst [Va06]. Dieser Ansatz bietet ein auf die Entwicklungsforschung angepasstes mehrstufiges Verfahren.

2 Live-Coding

Das Programmieren und Improvisieren in Echtzeit von elektronischer Musik und audiovisueller Kunst wird als Live-Coding bezeichnet [Ro15; Co03; BS09]. Als erster international bekannter Verbund von Künstlerinnen und Künstlern trat “Toplap”⁴ 2004 in Erscheinung. Bekannte Künstler und Wissenschaftler dieser Szene wie Alex McLean, Julian Rohrhuber und andere verfolgten mit diesem Zusammenschluss das Ziel, Live-Coding bekannter zu machen, mit anderen Künstlern und Künstlerinnen in Austausch zu treten und diese Form der Kunst wissenschaftlich zu erforschen und künstlerisch zu reflektieren. Mittlerweile hat sich die Live-Coding-Community international stark vergrößert und versucht, auch außerhalb des akademischen Kontextes ein breiteres Publikum anzusprechen. Auf sogenannten “Algoraves” soll die typische Konzertsaalatmosphäre aufgebrochen werden und gemeinsam zu algorithmisch erzeugter Musik getanzt werden [CM14].

Die Autorinnen erwarten, dass Live-Coding einen neuen Zugang zum Programmieren, Komponieren, Performen und Rezipieren von Musik schafft. Erste Veröffentlichungen zum Musikprogrammieren mit Sonic Pi [Bu14; Aa16] sowie die Ergebnisse aus den ersten Feldversuchen im Rahmen dieser Studie bestätigen diese Annahme. Schülerinnen und Schüler können den Computer neuartig als kreatives Instrument erleben und dabei ihre Konzepte von Musik, Klang, Musiker_in und Programmierer_in überdenken und neu ordnen [Gu13, S.136]. Auch Lernende ohne musikalische Vorbildung haben die Möglichkeit, einen Einstieg in die Musik zu finden. Schüler_innen mit Vorbehalten gegenüber der Informatik bekommen die Möglichkeit eines neuen “Türöffners” zur Informatik.

3 Entwicklung der Unterrichtseinheit

3.1 Planungsvoraussetzungen

Die Erprobung der Unterrichtseinheit wird im Rahmen des regulären Musikunterrichts durchgeführt. Dies ist zum einen darin begründet, dass es keinen Pflichtunterricht in Informatik und vielfach auch kein freiwilliges Angebot an den Schulen gibt, den Autorinnen aber das Erreichen aller Schüler_innen wichtig ist. Zum anderen schafft der Kontext des Musikunterrichts einen anderen Zugang zur Informatik, der eine große Offenheit insbesondere bei den Mädchen erkennen lässt.

Die Erprobung findet im Teamteaching statt. Grundsätzlich ist die Einheit aber so konzipiert, dass sie auch von musikaffinen Informatiklehrkräften unterrichtet werden kann, da der inhaltliche Fokus auf informatischen Konzepten liegt und die Musik hier hauptsächlich als Kontext dient.

⁴ <https://toplap.org> [07.06.2017]

Der inhaltlichen Planung voran ging eine Analyse der Musik-, Physik- und Informatikbildungspläne im Forschungsumfeld. Die Projektschulen sind zunächst Gymnasien und Realschulen in Baden-Württemberg, sodass für die ersten drei Durchführungen die hier gültigen Bildungspläne betrachtet wurden. Die Entwicklung der Einheit wurde im Dialog mit Lehrkräften der drei Fächer erarbeitet.

3.2 Vorwissen und fachliche Ziele

Im Bereich der Informatik wird keinerlei Vorwissen vorausgesetzt, da es zumindest an den meisten Schulen keinen Pflichtunterricht in der Unter- und Mittelstufe gibt. Die zentralen Informatikkonzepte, die in der Unterrichtseinheit behandelt werden sollen, sind Wiederholungen, Parameter, Variablen sowie einfache Datenstrukturen. Insgesamt soll bei den Schüler_innen das Bild von Programmierung als ein kreativer Prozess neu verankert und somit das Interesse an der Informatik geweckt werden. Der Umgang mit dem Raspberry Pi ist innerhalb dieser Einheit den affektiven Lernzielen zuzuordnen, da das Ziel ausschließlich in der Nutzung für Sonic Pi und dem Abbau der Vorbehalte gegenüber alternativen Betriebssystemen liegt. Hierfür sind zunächst keine tieferen fachlichen Kenntnisse über die Plattform notwendig. Eine nähere Behandlung des Systems im Nachgang der Einheit ist sicher sinnvoll, aber nicht Teil dieser Studie.

Im Physikunterricht wird im Anfangsunterricht das Thema Akustik eingeführt. Der Aufbau einer Schwingung ist den Schüler_innen daher bekannt, ebenso Fachbegriffe wie Amplitude und Frequenz sowie deren Bedeutung. In Sonic Pi sind dies zentrale Parameter, sodass das Vorwissen aus dem Physikunterricht zum einen hilfreich ist, zum anderen aber auch wiederholt und angewendet wird und so zu einer Festigung führt. Gleichzeitig bietet dies den Schüler_innen eine konkrete und motivierende Anwendung des erlernten Physikwissens.

Im Musikunterricht wurden in der Unter- und Mittelstufe die klassische Musik und ihre Formen und Strukturen behandelt, ebenfalls der Aufbau von Popsongs sowie Blues- und Jazz-Stücken. Grundzüge der Computermusik und die MIDI-Notation werden zwar thematisiert, diese Inhalte sollten jedoch im Rahmen der Unterrichtseinheit aufgefrischt werden. Ebenfalls sollte nochmals der Unterschied zwischen elektronisch verstärkten und elektronischen Musikinstrumenten thematisiert werden. Die Schüler_innen bringen außerdem Vorwissen aus dem Bereich der Harmonielehre und Rhythmik mit. Ebenfalls haben sie gelernt, über Musik zu sprechen und Klänge zu beschreiben. Innerhalb der Sonic Pi Einheit werden diese Kenntnisse angewendet. Inhaltlich neu ist das Mittel des Live-Coding.

3.3 Aufbau der Unterrichtseinheit

Die Unterrichtseinheit ist für fünf Doppelstunden konzipiert. Sie gliedert sich in einen festen Einführungsteil über zwei Doppelstunden, einen flexiblen Erarbeitungs- und Projektteil, der auch sehr gut auf einen längeren Zeitraum ausgedehnt werden kann und

einen Präsentations- und Reflexionsteil. Neben dieser linearen Struktur ist auch ein iterativer Aufbau mit mehreren Projektphasen sinnvoll, sofern mehr Unterrichtszeit zur Verfügung steht.

Den Schüler_innen wird freigestellt, Sonic Pi auf ihrem Rechner zu Hause zu installieren. Die Hausaufgaben wurden bewusst so gestaltet, dass diese unplugged bearbeitet werden und die Arbeit mit Sonic Pi in der Schule vorbereiten. Dies soll zum einen sicherstellen, dass Jugendliche ohne entsprechenden Zugang zu einem Rechner nicht benachteiligt werden, zum anderen auch den im Anfangsunterricht schwierigen Umgang mit Fehlern und Debugging in eine Umgebung verlegen, in der direkter Zugriff auf Hilfe durch die Lehrkraft möglich ist.

Die erste Stunde dient der Einordnung der Einheit in die historische Entwicklung der Musik und Informatik sowie dem ersten Kontakt mit unplugged Musikprogrammierung in Pseudocode. Der unplugged-Teil ist wichtig für einen niederschweligen Einstieg in die textuelle Programmierung und die spätere Umsetzung in Sonic Pi, da die Schüler_innen keinerlei Vorwissen aus der Programmierung mitbringen. In den Materialien⁵ wird daher auch großer Wert auf die farblichen und graphischen Entsprechungen der Arbeitsblätter im unplugged-Teil zu der späteren Programmierumgebung gelegt. In der zweiten Doppelstunde werden eng angelehnt an die unplugged Ergebnisse die ersten Schritte in Sonic Pi unternommen. Schüler_innen lernen die Kernkonzepte Codierung eines Tones, Wiederholung und Parameter sowie die Kommentierung ihres Codes kennen. Ebenso findet hier der in der Regel erste Umgang mit dem Raspberry Pi statt, der aber intuitiv und zweckorientiert erfolgt und keinen eigenen Unterrichtsgegenstand darstellt. In den folgenden Stunden werden dann in Teams eigene Stücke erstellt, Methoden, Variablen, Datenstrukturen und Zufallsbedingungen eingeführt. Wichtig ist bei der Teamarbeit die Steuerung der Programmierfähigkeit. Hier bietet sich Pair Programming an, um sicherzustellen, dass beide Teammitglieder aktiv mit der Umgebung und Programmiersprache arbeiten. Über Cheatsheets und gestufte Hilfen zu einzelnen Programmier-elementen in Sonic Pi stehen binnendifferenzierende Materialien zur Verfügung. Die letzte Sitzung dient der Präsentation des eigenen Stückes als Live-Coding Performance, einer inhaltlichen Diskussion der eingesetzten Konzepte sowie einer kurzen Reflexion über die Unterrichtseinheit aus Sicht der Schüler_innen.

4 Erste Durchführungsphase und Erkenntnisse

Die erste Durchführungsphase fand an einem Gymnasium am Ende der achten Klasse statt. Die Schüler_innen hatten großenteils keine informatische Vorbildung, einige hatten

⁵ Sämtliche Unterrichts- und Planungsmaterialien stehen zum Zeitpunkt der Konferenz als Entwicklungsfassung zur Verfügung und werden im Rahmen des Workshops ([AD17b], ebenfalls in diesem Band) vorgestellt. Nach Abschluss des Forschungsvorhabens werden die Unterrichtsmaterialien zusammen mit Stundenverlaufsplänen und anderen Handlungsempfehlungen veröffentlicht.

in Klasse fünf erste Programmiererfahrungen mit Scratch gemacht, seitdem aber keinen weiteren Kontakt mit Informatik in der Schule. Viele Jugendliche brachten musikalisches Vorwissen aus Instrumentalunterricht mit, das über die Inhalte des schulischen Musikunterrichts hinausging. Die Musiklehrerin hat die Entwicklung der Einheit aus musikalischer Sicht mit begleitet. Die Durchführung lag größtenteils bei der Informatiklehrkraft und wurde an einzelnen Stellen von der Musiklehrerin unterstützt.

4.1 Beobachtungen

Die Einheit lief für alle Beteiligten vielversprechend ab. Die Schüler_innen waren zunächst vor allem neugierig und gingen offen auf das Thema zu. Der Einstieg in die Programmierung fiel den meisten Schüler_innen leicht und von Beginn der Arbeit in Sonic Pi an zeigten sie sich hoch motiviert. Über die im Unterricht explizit vorgestellten Fachkonzepte hinaus stellten sie Fragen zur Umsetzung ihrer Ideen in Sonic Pi und recherchierten teilweise selbständig Möglichkeiten, um ihre Ideen in der Programmiersprache auszudrücken. Hierbei stellte das deutschsprachige Tutorial, das innerhalb Sonic Pi zur Verfügung steht, eine hervorragende Unterstützung dar. Die Schüler_innen haben während ihrer Programmierung ausnahmslos die Pausen missachtet und zum Teil zu Hause ihre Stücke freiwillig weiter entwickelt. In der Abschlusspräsentation wurden teilweise sehr komplexe Stücke als Ergebnis⁶ vorgestellt und als Live-Coding präsentiert.

Die Rückmeldungen der Schüler_innen und Lehrer_innen bestätigen die insgesamt sehr vielversprechenden Beobachtungen der Lehrkräfte und Autorinnen. Das Feedback zur ersten Doppelstunde fiel durchweg positiv aus. Der unplugged Einstieg als Methode war für die Lerngruppe neu, wurde aber gut angenommen und von Schüler_innen und Lehrkräften als sehr positiv und hilfreich bewertet. Die eigenen Stücke bzw. kurzen Performances der Schülerinnen und Schüler und deren Präsentation stellten den Abschluss der Projekteinheit dar. Die Ergebnisse übertrafen die Erwartungen: die künstlerische Vielfalt, die Komplexität der Stücke und auch die Programmierfähigkeiten wurden auf eindruckliche Weise dargestellt.

In den Interviews, die im Nachgang der Einheit durchgeführt wurden, wünschten sich die Schüler_innen eine klarere Zielsetzung und äußerten, schneller mit einem eigenen Stück anfangen zu wollen.

4.2 Erwartete und tatsächliche Schwierigkeiten

Viele Bedenken erwiesen sich als unbegründet. So wurde bspw. der Unterschied zwischen traditioneller Notation und Sonic Pi-Notation sehr schnell verstanden und stellte im Umgang mit Sonic Pi kein Problem dar.

⁶ Ergebnisse aus den Durchführungen werden im Rahmen des Workshops ([AD17b], ebenfalls in diesem Band) sowie im Vortrag vorgestellt.

Als problematisch erwies sich die Stabilität der Software auf Windows-Rechnern. Neben Abstürzen des Ruby-Interpreters kam es vor, dass die Audioausgabe nicht funktionierte und mehrere Dienste über den Taskmanager beendet werden mussten. Diese ist zwar mittlerweile vom Sonic Pi Entwicklerteam verbessert worden, dennoch nahmen die Autorinnen diese Erfahrung zum Anlass, auf Raspberry Pi umzusteigen, da dies eine schulunabhängige Evaluation ermöglicht und sie sich darüber hinaus den Abbau von Berührungsängsten mit anderen Plattformen und Betriebssystemen bei den Lernenden erhoffen.

Ein großes Problem stellten für alle Schüler_innen die Fehlermeldungen dar. Diese sind nicht kindgerecht dargestellt, sondern werden direkt von der zugrundeliegenden Sprache Ruby gesandt. Die Meisten gaben an, nach der Einheit zu Hause nicht weiter mit Sonic Pi arbeiten zu wollen, da sie sich nicht befähigt fühlten, selbst Fehlermeldungen zu lesen und den Programmcode entsprechend zu ändern. Dies ist insbesondere angesichts der hohen Motivation innerhalb der Einheit von Bedeutung: Das Interesse ist da, die Umsetzung scheitert aber an der Hürde des Debugging. Hier soll in nachfolgenden Durchführungen unterstützt werden, indem häufig auftretende Fehlermeldungen evaluiert und gezielte Erklärungen und Lösungshinweise in den Unterricht und die Materialien eingearbeitet werden. Insgesamt muss der Umgang mit Fehlern in Code und Software als Unterrichtsinhalt und Lernziel stärker berücksichtigt werden.

Einigen Schüler_innen war die Aufgabe, ohne einen programmatischen Inhalt frei zu komponieren, zu komplex. Hier wird es in den nächsten Durchführungen ein konkretes Projektthema vorgegeben. Dieses kann je nach Lerngruppe auch als Angebot genutzt werden, das eigene Ideen weiterhin zulässt. Die Vertonung eines Kurzfilms gibt daher als Rahmen für die zweite Erprobung ein klareres Ziel und eine klare Aufgabenstellung vor.

Da die Schüler_innen meist eine rein klassische Musikausbildung in der Schule erhalten, möchten sie natürlich ihnen bekannte musikalische Strukturen umsetzen. Fragen, wie eine klassische Songstruktur, der zeitlich versetzte Einsatz verschiedener Instrumente, ein Crescendo o.ä. in Sonic Pi umzusetzen sind, tauchen während der Arbeit an der eigenen Komposition auf. Hier müssen von der Lehrkraft im Vorfeld selbst einige kleine Kompositionen angefertigt und die Beispiele, die in Sonic Pi abrufbar sind, bearbeitet werden, um die Schülerfragen gezielt beantworten zu können. Dieser Punkt wird ebenfalls im Zuge der weiteren Evaluation berücksichtigt und mit Material unterstützend vorbereitet werden.

Aus Sicht der Informatik kann mit Sonic Pi sehr einfach ein intuitiver Umgang mit Schleifen, Methoden, Parametern und Variablen erreicht und damit der Einstieg in eine textuelle Programmiersprache niederschwellig erreicht werden. Die Umgebung differenziert beispielsweise sehr gut zwischen Variablen und Konstanten, setzt Syntax Highlighting reduziert und zielführend ein und wird von den Lernenden problemlos bedient. Das direkte auditive Feedback stellt sich im Unterricht als hoch motivierend dar. Im Vergleich zu den Erfahrungen mit visuellem Feedback beispielsweise in Greenfoot

zeigte sich eine höhere Motivation insbesondere bei den Mädchen.

Problematisch ist die Umsetzung des Konzeptes Verzweigung. In Sonic Pi gibt es nur zufallsbedingte Verzweigungen, sodass hier einer Fehlvorstellung im Rahmen des Unterrichts entgegengewirkt werden muss. Dies ist keine Schwachstelle der Programmiersprache, sondern vielmehr der Tatsache geschuldet, dass das Ziel die Ausgabe von Musik ist, was Sensorik in den Hintergrund stellt.

5 Ausblick

In der zweiten und dritten Durchführungsphase wird das überarbeitete Konzept erprobt, verbessert und angepasst. Folgende Änderungen wurden für die erneute Durchführung auf Basis der Ergebnisse aus teilnehmender Beobachtung und Interviews vorgenommen:

- Sonic Pi stürzte unter Windows leider häufig ab. Dieses Problem ist den Entwicklern bekannt und wurde versucht in der aktuellsten Sonic Pi-Version zu verbessern. Wir empfehlen dennoch eine Nutzung von Sonic Pi mit Raspberry Pi (für die es entwickelt wurde). Ein positiver Nebeneffekt ist zudem der Kontakt mit einer alternativen Plattform. Die Schülerinnen und Schüler können somit ganz nach dem Motto “Technik zum Anfassen” Berührungängste mit unbekannter Hard- und Software, insbesondere auch mit Linux als Betriebssystem, abbauen.
- Für Klassen, in denen viel Unruhe herrscht, wurde das Konzept dahingehend überarbeitet, dass der Theorieinput deutlich kürzer und auf mehrere Sitzungen verteilt wurde.
- Das Projektziel wird zukünftig enger vorgegeben. Einige Schüler_innen waren mit der Aufgabenstellung, frei zu komponieren, überfordert. Daher wird als Rahmen zukünftig die gemeinsame Vertonung eines Kurzfilms oder eines Bildes vorgegeben. Denkbar wäre ebenfalls die Vertonung von Jahreszeiten oder Wochentagen. Wichtig ist hierbei, dass zum Abschluss der Unterrichtseinheit eine gemeinsame Reflexion über die entstandenen Werke geführt wird. Ebenfalls werden zukünftig Kompositionshilfen in Form von Genrevorlagen an die Schüler_innen bei Bedarf ausgegeben. Diese Vorlagen können Beispiele für typische rhythmische Patterns, Melodieführungen und Basslinien eines bestimmten Musikgenres o.ä. enthalten.
- Die Übungsphase mit traditionellen, einfachen Liedern wurde verkürzt, so dass mehr Zeit für das freiere Ausprobieren und Komponieren bleibt. Im Zuge dessen wurde die Unterrichtseinheit um eine Doppelstunde (also insgesamt 6 Doppelstunden) verlängert.
- Um Live-Coding und die damit verbundene Performance für Schüler auditiv und visuell greifbarer zu machen, werden innerhalb der Einheit künftig mehrere kurze Videoausschnitte eingebaut.

- Das Debugging wird als Unterrichtsgegenstand stärker berücksichtigt. Die häufigsten Fehlermeldungen werden im Rahmen der nächsten Durchführung erhoben und auf dieser Basis unterstützendes Material erarbeitet.

6 Fazit

Zusammenfassend kann gesagt werden, dass die Ergebnisse der ersten Erprobungen des Zugangs zur Informatik über die Musikprogrammierung vielversprechend sind. Die Schülerinnen und Schüler arbeiteten mit großer Begeisterung und hoher Motivation in Sonic Pi an ihren eigenen Kompositionen. Sie suchten eigenständig nach eleganten Lösungen für musikalische Elemente und nutzen die erlernten Programmierelemente sinnvoll und zielführend. Die Arbeitsergebnisse waren auch aus künstlerischer Sicht komplex und vielseitig. Die Autorinnen sehen in Sonic Pi eine reizvolle Alternative zum Einstieg in die textuelle Programmierung, der nach ersten Beobachtungen insbesondere Mädchen anspricht und für Informatik begeistern kann.

Literaturverzeichnis

- [Aa16] Aaron, S., Blackwell, A.F.; Burnard, P.: The development of Sonic Pi and its use in educational partnerships: co-creating pedagogies for learning computer programming. *Journal of Music, Technology and Education* 9(1)/16, S.5-94, 2016.
- [AD17b] Alzate Romero, E.; Dietrich, L.: Workshop: Musikprogrammierung mit Sonic Pi. In (Ira Diethelm, Hrsg.): *Informatische Bildung zum Verstehen und Gestalten der digitalen Welt*, *Lecture Notes in Informatics (LNI)*, Gesellschaft für Informatik, Bonn 2017.
- [BS09] Brown, A. R.; Sorensen, A.: Interacting with Generative Music through Live Coding. *Contemporary Music Review*, 28(1), S.17–29, 2009.
- [Bu14] Burnard, P. et.al. (Hrsg): *Sonic Pi. Live and Coding*. Research Report, 2014.
- [CM14] Collins, N.; McLean, A.: Algorave: A Survey of the History, Aesthetics and Technology of Live Performance of Algorithmic Electronic Dance Music. In (Baptiste Caramiaux, K. et.al. Hrsg.): *Proc. International Conference on New Interfaces for Musical Expression*, S.355–358, 2014.
- [Co03] Collins, N.; McLean, A.; Rohrhuber, J.; Ward, A.: Live coding in laptop performance. In: *Organised Sound*, 8(3)/03, S.321–329, 2003.
- [GI08] Gesellschaft für Informatik (GI) (Hrsg.): *Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I*. In: *LOG IN* 28/08, Nr.150/151/08, <http://informatikstandards.de>, Stand 07.06.2017.
- [Gu13] Guzdial, M.: Live-Coding in Education. In (Blackwell, A.; McLean, A.; Noble, J.; Rohrhuber, J., Hrsg.): *Collaboration and learning through live coding (Dagstuhl Seminar 13382)*. *Dagstuhl Reports*, 3/13, S. 135-136, 2013.

- [IWK16] Institut der deutschen Wirtschaft Köln (Hrsg.) (2016): MINT-Herbstreport 2016, <https://www.iwkoeln.de/studien/gutachten/beitrag/christina-anger-oliver-koppel-axel-pluenecke-mint-herbstreport-2016-bedeutung-und-chancen-der-zuwanderung-315792>, Stand 20.03.2017.
- [Me16] Medienpädagogischer Forschungsverbund Südwest (Hrsg.) (2017): KIM-Studie 2016. Kindheit, Internet, Medien. Basisstudie zum Medienumgang 6- bis 13-Jähriger in Deutschland, <https://www.mpfs.de/studien/kim-studie/2016/>, Stand 07.06.2017.
- [Ro15] Rohrhuber, J.: Lehre als Forschung: Grundlagen der Musikinformatik im künstlerischen Kontext. Zeitschrift für Hochschulentwicklung, 10(1)/15, S.79-91, 2015.
- [Sc12] Schinzel, B.: Geschlechtergerechte Informatik-Ausbildung an Universitäten. In (Kampshoff, M.; Wiepcke, C. Hrsg.): Handbuch Geschlechterforschung und Fachdidaktik. VS Verlag, Wiesbaden, S.331-344, 2012.
- [Va06] Van den Akker, J. et.al. (Hrsg.): Educational Design Research, Routledge, London, 2006.
- [Ze07] Zendler, A., Spannagel, C. & Klaut, D.: Zentrale Prozesse im Informatikunterricht: eine empirische Grundlegung. In: Notes on Educational Informatics - Section A: Concepts and Techniques, 3(1)/07, S.1-19, 2007.
- [ZS06] Zendler, A. & Spannagel, C.: Zentrale Konzepte im Informatikunterricht. Eine empirische Grundlegung. In: Notes on Educational Informatics — Section A: Concepts and Techniques, 2(1)/06, S.1–21, 2006.

Entwicklung eines HTML-Editors unter didaktischen Gesichtspunkten

Dominik Heun¹

Abstract: Im Kerncurriculum Informatik des Landes Hessen für die gymnasiale Oberstufe wird die Umsetzung eines HTML-Projekts gefordert. Dabei stellt sich regelmäßig die Frage, mit welchen Werkzeugen und Editoren eine HTML-Seite geschrieben werden soll. Viele frei verfügbare Editoren stellen weitreichende Funktionen zur Verfügung, die in vielen Fällen zu weit gehen oder Informatikdidaktisch interessante Aspekte außen vorlassen.

In diesem Paper werden didaktische Anforderungen an einen Editor formuliert, die anschließend mit einem selbst entwickelten Editor erfüllt werden sollen. Es werden Grundfunktionen des Editors (HTML, CSS, JavaScript, Dokumentenbaum) vorgestellt, Hilfen in der Erstellung von HTML-Dokumenten diskutiert und Erfahrungen aus dem Einsatz im Unterricht der Einführungsphase vorgestellt.

Keywords: Gymnasium, Oberstufe, Einführungsphase, HTML, CSS, Webseiten

1 Einleitung

Innerhalb der Einführungsphase (zwei Jahre vor dem Abitur) können Schülerinnen und Schüler an Gymnasien in Hessen das Fach Informatik wählen. Darin erhalten sie nach dem Kerncurriculum der gymnasialen Oberstufe² eine informatische Grundbildung in verschiedenen Themenbereichen, die im ersten Halbjahr um das Thema Internet aufgebaut ist. Nach der Behandlung des Themenbereichs Internetprotokolle wird ein HTML-Protokoll angesetzt. Für das HTML-Projekt werden drei grundlegende Inhalte beschrieben: die Hypertext Markup Language selbst (HTML), die Struktur von HTML-Dokumenten sowie Cascading Style Sheets (CSS).

2 Kriterien für einen HTML-Editor

Für die Lehrenden ist zu Beginn der Unterrichtsreihe die Auswahl eines geeigneten Editors nötig, der von den Schülerinnen und Schülern genutzt werden soll. Diese Suche gestaltet sich schwierig, da einige Editoren wie bspw. die Standard-Editoren von Windows oder Mac reine Texteditoren sind und keine Unterstützung in Bezug auf HTML-Dokumente geben. Phase 5³ bietet genauso wie Notepad++⁴ eine rudimentäre

¹ Justus-Liebig-Schule Darmstadt, Julius-Reiber-Straße 3, 64293 Darmstadt, heu@lio-darmstadt.de

² [Hes15]

³ Für Schulen und Privatanwender kostenlos unter <http://www.phase5.info> (nur Windows)

⁴ Kostenlos verfügbar unter <https://notepad-plus-plus.org>

Unterstützung durch Hervorhebung der Syntax (Syntax-Highlighting) und automatisches Einrücken zur Strukturierung. Der Funktionsumfang dieser Editoren ist nicht nur auf HTML eingeschränkt, weswegen die Unterstützungen auch nicht HTML-spezifisch sind. Ein vollständiger „What you see is what you get“-Editor (WYSIWYG), der ohne Kenntnis von HTML genutzt werden kann, ist nicht im Sinne des Kerncurriculums.

Welche Unterstützungen sind für den Informatikunterricht mit HTML wünschenswert? Dazu ist ein Blick auf Schwierigkeiten von Schülerinnen und Schülern im Bereich von HTML hilfreich. Im bisherigen Unterricht des Autors zeigten sich unter anderem folgenden Probleme der Schülerinnen und Schüler (die nummerierte Aufzählung gibt keine Gewichtung vor, sondern dient dem späteren Referenzieren):

1. Korrekte Syntax von HTML-Elementen
2. Bezug zwischen HTML-Quelltext und entstehendem Dokument bleibt unklar
3. Arbeit in drei Fenstern zeitgleich (HTML, CSS und Browser)
4. Korrekte Verknüpfung von HTML- und CSS-Dateien
5. Identifizierung von HTML-Knoten, die durch CSS-Regeln formatiert werden
6. Zusammenhang zwischen Inhalt, Struktur und Aussehen

Des Weiteren gestalten sich gestufte Hilfen schwierig, da diese in einem erneuten (evtl. analogen) Format angeboten werden und damit ein weiterer Arbeitsbereich für die Schülerinnen und Schüler entsteht. Diese Vielzahl kann einige Lernende überfordern und sollte deshalb von einem HTML-Editor besser gelöst werden.

Aus dem ersten Punkt wird deutlich, dass ein Syntax-Highlighting eine mögliche Unterstützung darstellt. Dies unterstützt Lernende aber lediglich dann, wenn diese schon eine Ahnung davon haben, wie sie ein HTML-Element schreiben können.

Der zweite und dritte Punkt bedingen sich gegenseitig. Durch die Arbeit in vielen verschiedenen Fenstern oder Tabs sind die Auswirkungen von Änderungen an den Dokumenten nicht direkt sichtbar, sondern werden erst über Umwege sichtbar gemacht. Dies stört den Arbeitsfluss der Lernenden durch zusätzliche Routinen, die bei jeder Änderung ausgeführt werden müssen.

Punkt vier stellt gerade zu Beginn des Themenbereichs ein großes Problem dar, das viele Schülerinnen und Schüler vor eine zusätzliche Hürde stellt. Die Lernenden sollten am Ende der Unterrichtsreihe selbst eine solche Verknüpfung herstellen können, für die erste Begegnung mit Unterrichtsgegenstand ist dies jedoch ein vermeidbares Hindernis.

Die beiden letzten Punkte hängen ebenfalls zusammen, da die Identifizierung von HTML-Knoten, die durch eine CSS-Regel verändert werden, über die Struktur erfolgt, die den Lernenden jedoch nicht zu jeder Stelle klar ist. Zwar können sie einen Strukturbaum zu ihrem HTML-Dokument erstellen, nach einer Änderung am Dokument

muss dieser aber überarbeitet werden, so dass der Strukturbaum parallel weitergepflegt werden muss.

Im Zuge der Differenzierung wäre es ebenfalls hilfreich, wenn der HTML-Editor direkt die Aufgabenstellung und zusätzliche Hilfen anzeigen könnte, damit alle nötigen Arbeitsmaterialien an einem Platz sind.

3 Vorbilder

3.1 Codefights

Auf der Online-Plattform codefights.com können Nutzer sich anmelden und verschiedene tägliche oder wöchentliche Herausforderungen lösen.

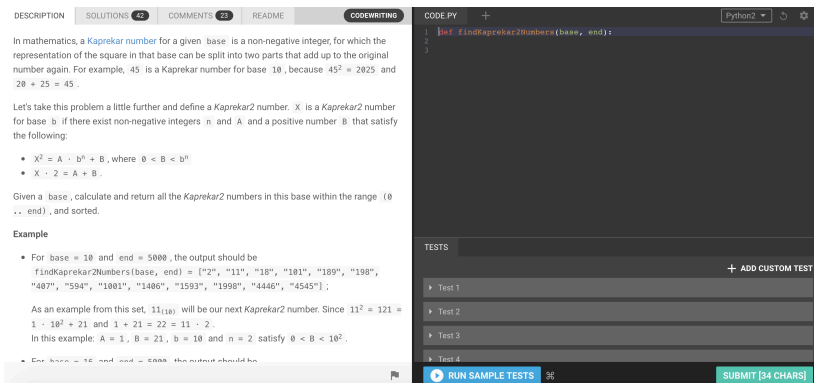


Abbildung 1: Beispielhafte Herausforderung bei Codefights

Dabei wird im Browser die Aufgabenstellung und ein Code-Editor angezeigt. Die Herausforderung lässt sich in einer Sprache der Wahl lösen und kompilieren. Glaubt man die Herausforderung gelöst zu haben, kann man Tests laufen lassen, deren Ergebnis angezeigt wird. Dieses HTML-fremde Beispiel zeigt einen Ansatz für Editoren innerhalb von Web-Browsern auf.

Dieser Ansatz bietet einige Vorteile gegenüber der einfachen Editorlösung. Eine Aufgabenstellung wird dauerhaft angezeigt, die Programmierung erfolgt direkt im gleichen Fenster. Es werden Tests angezeigt, mit denen Lösungen überprüft werden können. Zusätzlich bietet die Umsetzung in HTML und JavaScript einen Vorteil für die Plattformunabhängigkeit des Editors.

3.2 Mozilla Thimble

Verfolgt man diesen Ansatz weiter, wird man auf das Projekt Thimble von Mozilla Firefox aufmerksam, das einen ähnlichen Ansatz verfolgt und für HTML umsetzt.

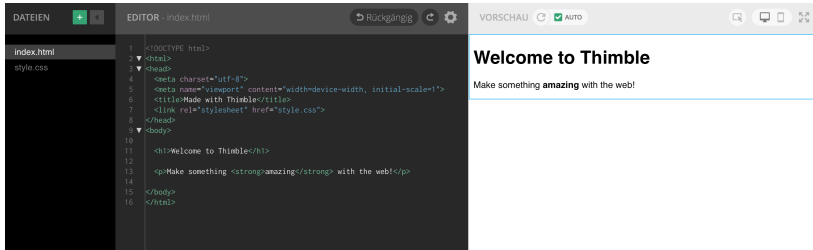


Abbildung 2: Mozilla Thimble

In einem Editor (Bereich in der Mitte) können HTML-Elemente eingegeben werden, die direkt in die Vorschau auf der rechten Seite übernommen werden. Man kann die Entstehung des Dokuments verfolgen und Auswirkungen von HTML-Elementen direkt beobachten. In der linken Liste kann zwischen HTML und CSS gewechselt werden. Auch innerhalb der CSS-Codes werden Auswirkungen unmittelbar in der Vorschau angezeigt.

Thimble wirkt einigen der angesprochenen Probleme entgegen. Dennoch ist der Einsatz in der Schule nicht ohne Weiteres umsetzbar, da für die Nutzung ein Nutzerkonto jeder Schülerin und jedes Schülers vorhanden sein muss. Zusätzlich liegen die Daten der Lernenden auf einem Cloud-Server außerhalb Deutschlands. Beide Argumente konnten datenschutzrechtlich für die Schule nicht restlos geklärt werden, so dass diese Möglichkeit nicht in Betracht gezogen wurde.

4 Eigenentwicklung

4.1 Überblick

Aus den beiden vorgestellten Editoren ergab sich eine grobe Richtung, in die eine Eigenentwicklung gehen sollte. Basierend auf HTML und JavaScript wurde ein eigener HTML-Editor geschrieben.

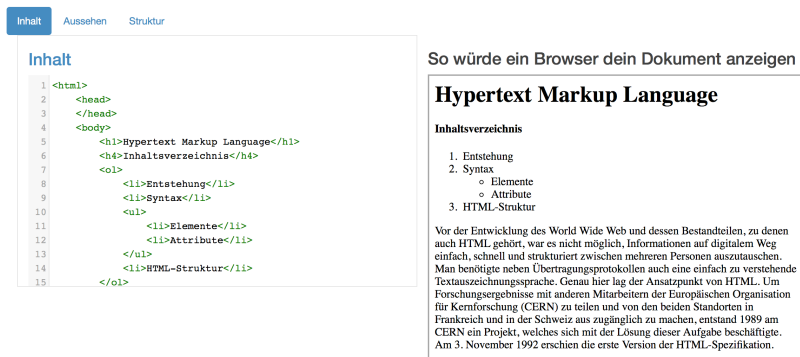


Abbildung 3: HTML-Editor

Die Anleihen zum Thimble sind erkennbar. Auch dieser Editor bietet die Aufteilung in Quelltext und Vorschau und genau wie Thimble werden Änderungen im HTML-Quelltext direkt in der Vorschau umgesetzt. Der HTML-Editor weist ein rudimentäres Syntax-Highlighting auf und macht auf nicht geschlossene HTML-Tags aufmerksam. Eine Syntax-Vervollständigung wird nicht zur Verfügung gestellt.

Oberhalb des Editors werden drei Tabs angezeigt, mit denen Schülerinnen und Schüler zwischen HTML und CSS wechseln können. Änderungen im CSS-Editor werden ebenfalls live in der Vorschau angepasst. Die Verknüpfung von HTML und CSS wird implizit vom Programm vorgenommen.

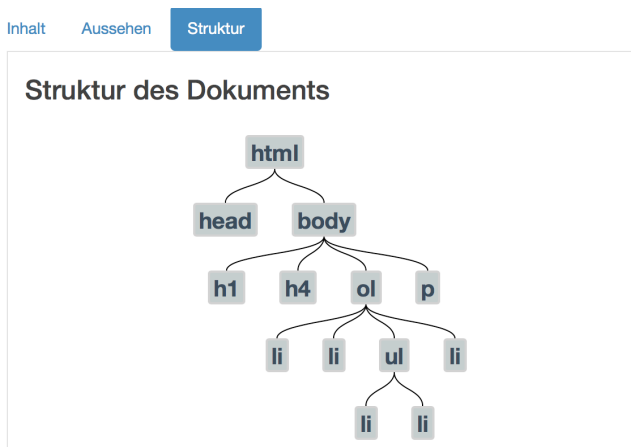


Abbildung 4: Strukturbaum des HTML-Dokuments

Der Editor bietet einen dritten Tab *Struktur*. Damit kann der aktuelle Strukturbaum des HTML-Dokuments angezeigt werden. Wird der HTML-Quelltext geändert, wird diese Änderung dynamisch am Strukturbaum vorgenommen.

4.2 Bereitstellen der Aufgabenstellung

Eine Aufgabenstellung kann den Schülerinnen und Schüler auf unterschiedliche Weise präsentiert werden. Die einfachste Variante ist das Einblenden oberhalb des Editors. So haben die Schülerinnen und Schüler den Editor, die Vorschau und die jeweilige Aufgabenstellung direkt auf einen Blick. Die verschiedenen Editoren können mit unterschiedlichen Quelltexten vorgelegt werden, um den Lernenden ein Template zum Arbeiten vorzugeben.

Die stärker leitende und damit für den Anfang besser geeignete Variante setzt auf kontextabhängige Aufgabenstellungen. Ein Beispiel: die Schülerinnen und Schüler erhalten zu Beginn ein leeres HTML-Dokument mit der übergreifenden Aufgabe, das HTML-Grundgerüst zu erstellen. Dazu erhalten sie den weiteren Hinweis einen öffnenden und einen schließenden Tag mit dem Namen `html` zu erstellen. Nachdem sie dies ausgeführt haben, wird der Schritt abgehakt und der nächste Schritt (Erstellung eines `head`-Elements) angezeigt. Somit wissen die Schülerinnen und Schüler jederzeit, was ihr nächster Schritt ist und können so zu einem vollständigen HTML-Grundgerüst gelangen.

Es ist ebenfalls ein Hybrid aus diesen beiden Varianten möglich, so dass die Schülerinnen und Schüler selbständig entscheiden können, welche Hilfen sie in Anspruch nehmen.

4.3 Bereitstellung von Hilfen

Um einem differenzierenden Unterricht gerecht zu werden, wurde der Editor erweitert, so dass gestufte Hilfen ermöglicht werden. Über verschiedene Buttons können die Schülerinnen und Schüler Hilfen zur Aufgabenstellung einblenden. Dies lässt sich ebenfalls an den Kontext anpassen, so dass den Schülerinnen und Schülern an verschiedenen Abschnitten ihrer Lösung auch verschiedene Hilfen angeboten werden.

Speziell für den Beginn der Behandlung von HTML ist das Grundgerüst zunächst eine Hürde. Deshalb lässt es sich ausschalten, so dass es für die Schülerinnen und Schüler unsichtbar ergänzt wird.

Ebenfalls für den Anfang kann eine Unterstützung bei der Erstellung von HTML-Elementen helfen. Dazu wurde ein Formular implementiert, das die Eigenschaften des gewünschten Elements abfragt und dann in den Quelltext einfügt.

Elementname

Attributname +
weitere
Attribut

Attributswert

Abbildung 5: Formular zur Erstellung von HTML-Elementen

Damit lassen sich ebenfalls die Fachbegriffe der Bestandteile eines HTML-Elements festigen. Dieses Formular ist auch für CSS-Regeln verfügbar.

Um die Anwendung von CSS-Regeln zu visualisieren, können Schülerinnen und Schüler den Strukturbaum verwenden. Um die angewandten Regeln besser deutlich zu machen, werden Veränderungen am Attribut *color* auf Wunsch im Strukturbaum angezeigt. Dies fördert das Verständnis des Zusammenhangs zwischen den Bereichen Inhalt, Struktur und Aussehen.

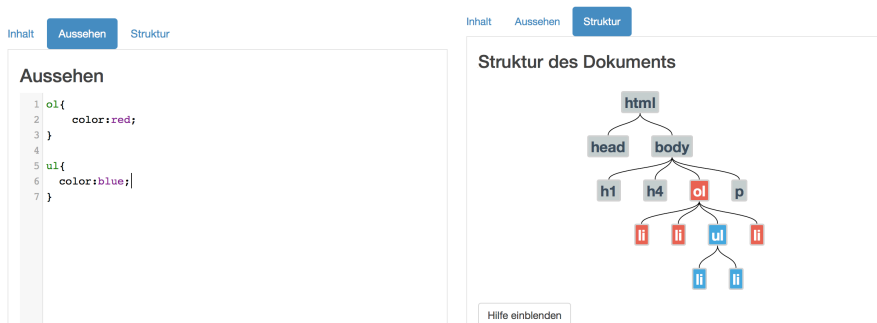


Abbildung 6: Visualisierung der CSS-Regeln am Strukturbaum

5 Einsatz im Unterricht

Der vorgestellte HTML-Editor wurde zu Beginn des Themenbereichs HTML-Projekt eingesetzt. Um ein generelles Verständnis für die Beschreibungssprache zu fördern, erhielten die Schülerinnen und Schüler den HTML-Editor ohne die Tabs Aussehen und Struktur. Innerhalb des HTML-Editors befand sich reiner Text eines Wikipedia-Artikels. Das HTML-Grundgerüst wurde automatisch vom Editor ergänzt, so dass die Lernenden direkt mit dem Text arbeiten konnten.

In einem Unterrichtsgespräch wurden die Formatierungen gemeinsam erarbeitet und direkt im Text umgesetzt. So konnten die Schülerinnen und Schüler den direkten Zusammenhang zwischen dem ursprünglichen Text und der Erweiterung durch die HTML-Tags erkennen. Ferner wurden die Auswirkungen der HTML-Elemente den Lernenden direkt deutlich und sie konnten es von sich aus beschreiben.

Nach diesem Einstieg wurde den Schülerinnen und Schülern erläutert, dass in HTML-Dokumenten ein Grundgerüst nötig sei. Dies sollten die Schülerinnen und Schüler über die in 4.3 beschriebene, kontextabhängige Hilfe erarbeiten.

Im nächsten Schritt wurde die Struktur von HTML-Dokumenten behandelt. Die Lernenden erhielten einen neuen Editor, bei dem der Tab Struktur freigeschaltet war. Über den Vergleich von HTML-Dokument und erstelltem Dokumentenbaum war es den Schülerinnen und Schülern möglich, Regeln zur Erstellung eines Dokumentenbaums aufzustellen. Die Auswirkungen von Veränderungen am HTML-Dokument konnten direkt im Dokumentenbaum untersucht werden.

In einem nächsten Schritt wurde der Tab Aussehen (CSS) freigeschaltet. Über die Vorbelegung des CSS-Editors mit einer Regel konnten die Schülerinnen und Schüler die Folgen der CSS-Regel direkt im Dokument beobachten. Analog entwickelten sie eigene Regeln, die sich zunächst nur auf Tag-Namen bezogen. Der Bezug zwischen Dokumentenbaum und den CSS-Regeln konnte durch die eingeblendeten Hilfen hergestellt werden. Die Schülerinnen und Schüler konnten durch die Beobachtungen, die sich immer auf ihre eigenen Beispiele bezogen, selbst formulieren, wann eine CSS-Regel auf ein HTML-Element angewandt wird.

Aufgabenstellung

Du hast ein neues Dokument erhalten. Wieder gibt es ein Layout-Dokument. Über den Menüpunkt "Struktur" kannst du dir den Strukturbaum einblenden.

Notiere dir auf einem Blatt:

1. Begründe anhand des Strukturbaums, warum das Dokument so formatiert wurde.
2. Schreibe einen Befehl, der alle ul-Listen grün färbt. Interpretiere das Ergebnis anhand des Strukturbaums.
3. Formuliere einen Merksatz, der die Regeln für das Anpassen von HTML-Elementen beschreibt. Du kannst dir Formulierungshilfen einblenden.
4. ***Füge der Formatierung für ol-Listen die Zeile "font-size:14pt;" hinzu und interpretiere das Ergebnis anhand des Strukturbaums.***

The screenshot shows a web editor with a menu bar containing 'Inhalt', 'Aussehen', and 'Struktur'. Below the menu is a code editor with the following HTML code:

```

4      <body>
5      <h1>Hypertext Markup Language</h1>
6      <h4>Inhaltsverzeichnis</h4>
7      <ol>
8          <li>Entstehung</li>
9          <li>Syntax</li>
10     </ol>
11     <ul>
12         <li>Elemente</li>
13         <li>Attribute</li>
14     </ul>
15     <li>HTML-Struktur</li>
16 </body>
17 <p> Vor der Entwicklung des World Wide Web und desse
18 </html>
    
```

To the right, a browser preview window titled 'So würde ein Browser dein Dokument anzeigen' displays the rendered HTML. The title is 'Hypertext Markup Language'. Below it is a section 'Inhaltsverzeichnis' containing a numbered list:

1. Entstehung
2. Syntax
 - Elemente
 - Attribute
3. HTML-Struktur

Below the list, the text from the code is rendered: 'Vor der Entwicklung des World Wide Web und dessen Bestandteilen, zu denen auch HTML gehört, war es nicht möglich, Informationen auf digitalem Weg einfach, schnell und strukturiert zwischen mehreren Personen auszutauschen. Man benötigte neben Übertragungsprotokollen auch eine einfach zu verstehende Textauszeichnungssprache. Genau hier lag der Ansatzpunkt von HTML. Um Forschungsergebnisse mit anderen Mitarbeitern der Europäischen Organisation für Kernforschung (CERN) zu teilen und von den beiden Standorten in Frankreich und in der Schweiz aus zugänglich zu machen, entstand 1989 am CERN ein Projekt, welches sich mit der Lösung dieser Aufgabe beschäftigte. Am 3. November 1992 erschien die erste Version der HTML-Spezifikation.'

Abbildung 7: Komplette Lernumgebung

Nachdem die Grundlagen mit dem Editor gelegt wurden, sollten die Schülerinnen und Schüler in Gruppen eine eigene HTML-Seite erstellen. Dazu wurden die traditionellen

Editoren (siehe 2) zur Verfügung gestellt, die von einigen Lernenden genutzt wurden. Ungefähr die Hälfte der Lerngruppe verblieb aber in dem vorgestellten Editor und kopierte anschließend den Text aus den Dokumenten und fügte ihn im Windows-Editor ein.

Die Schülerinnen und Schüler hatten keine Schwierigkeiten mit dem Editor zu arbeiten und fanden sich in den verschiedenen Stufen gut zurecht. Es traten jedoch Probleme mit dem Internet-Explorer auf, die während der Durchführung nicht behoben werden konnten, so dass auf einen anderen Browser zurückgegriffen werden musste. Die kontextabhängige Hilfe erwies sich ebenfalls als fehleranfällig, was aber behoben werden konnte. So zeigte die Hilfe fälschlicherweise nicht den Abschluss einer Zwischenaufgabe an, wenn mehrere Tags des gleichen Namens vorhanden waren. Des Weiteren trat einmal ein Absturz des Browsers auf, bei dem der bisherige Fortschritt der Schülerinnen und Schüler verloren war. Über die Cache-Möglichkeiten von HTML5 wurde ein Zwischenspeicher eingerichtet, der regelmäßig die Zwischenergebnisse der Schülerinnen und Schüler speichert, so dass die bisherige Arbeit nach einem Absturz verfügbar bleibt.

6 Ausblick

Der vorgestellte Editor stellt eine gute Möglichkeit dar, einen Editor im Unterricht zu verwenden, der die Schülerinnen und Schüler durch geeignete Maßnahmen beim Erlernen von HTML und CSS unterstützt. Er gibt für den Einstieg verschiedene Darstellungsformen, um einen explorativen Ansatz bei der Arbeit mit HTML zu ermöglichen. In der Erprobung zeigten sich die Einsatzfähigkeit im Unterricht sowie kleinere Probleme, die behoben wurden. Schon während des Einsatzes wurden Erweiterungen umgesetzt, wie bspw. ein weiterer Tab, in dem JavaScript geschrieben werden kann.

Für die Zukunft sind noch Erweiterungen des Editors geplant. So soll es eine Export-Funktion geben, die aus den aktuellen Inhalten der Editoren HTML- und CSS-Dateien erstellt und zum Download anbietet. Die Visualisierung der CSS-Regeln wird im Dokumentenbaum nur beim Attribut *color* angezeigt. Bei der Weiterentwicklung sollen alle Regeln über ein Mouse-Over angezeigt werden. Dies ermöglicht eine noch intensivere Umsetzung mit dem Zusammenhang zwischen CSS und der Struktur des HTML-Dokuments. Des Weiteren soll die Erstellung von Aufgaben für Lehrende vereinfacht werden. Aktuell wird der Editor direkt in der JavaScript-Programmierung von Aufgabe zu Aufgabe angepasst, was sehr zeitintensiv ist.

Der Autor schreibt ein Konfigurations-Tool, mit dem Aufgaben, sichtbare Tabs, Hilfen usw. festgelegt werden können und daraus eine HTML-Datei generiert wird, die den Editor entsprechend umsetzt. Speziell die kontextabhängigen Hilfen haben sich in der Erprobung als sehr hilfreich erwiesen, beanspruchen in der Vorbereitung aber sehr viel Zeit. Mit dem einfacheren Konfigurations-Tool würde diese Hilfe einfacher umsetzbar.

Diese Erweiterung würde es ebenfalls ermöglichen, dass dieser Editor von anderen Lehrenden verwendet werden kann und auf deren jeweilige Unterrichtssituation angepasst werden kann. Momentan erfordert die Anpassung eine intensive Einarbeitung in die Programmierung des Editors.

Literaturverzeichnis

[Hes15] Hessisches Kultusministerium: Kerncurriculum gymnasiale Oberstufe – Gymnasium. Informatik (2015).

Informatische Bildung in der Sekundarstufe I an nordrhein-westfälischen Schulen

Eine explorative Studie zum Einfluss der GI-Bildungsstandards in NRW

Johanna Borsch¹, Marco Thomas²

Abstract: Um fundiertere Informationen zur Situation einer informatischen Bildung in Nordrhein-Westfalen zu gewinnen, werden von unserem Arbeitsbereich seit 2013 vorwiegend quantitative Daten im Kontext von Informatikunterricht bei Schülern-, Lehrern- und Schulleitern erhoben sowie vergleichend ausgewertet. Unsere aktuelle Studie zur Sekundarstufe I ist auf das Angebot der Schulen zur informatischen Bildung und die Akzeptanz der GI-Bildungsstandards beziehungsweise die an den Standards orientierten neuen SI-Kernlehrpläne, aber auch den Fortbildungsbedarf, fokussiert. Wir erhielten Rückmeldungen von mehr als 200 Schulen. Im Rahmen des Vorhabens wurde erneut deutlich, wie schwierig sich die Durchführung universitärer Studien für den Schulbereich gestaltet.

Keywords: Informatik, Unterricht, Didaktik, Standards, Bildung, Sekundarstufe, Schule

1 Einleitung

In Ermangelung bundesweiter Bildungsstandards zum Fach Informatik hat die Gesellschaft für Informatik e. V. (GI) Empfehlungen [GI08] erarbeitet, die sich zunehmend als Standard für eine informatische Bildung und einen Informatikunterricht etablieren³. Mit den neuen Kernlehrplänen (KLP) zur Informatik für die Sekundarstufe I (SI) an Realschulen [KR15] und an Gesamtschulen/Sekundarschulen [KG15] für die Wahlpflichtbereiche hat Nordrhein-Westfalen (NRW) als eines der ersten Bundesländer die GI-Empfehlungen curricular nahezu vollständig verankert. Für die Gymnasien sind SI-Kernlehrpläne für den Wahlpflichtbereich in der Entwicklung⁴. Der Kernlehrplan Informatik für die Oberstufe ist ebenfalls stark an den GI-Bildungsstandards orientiert und für die Grundschulen werden in NRW diverse Konzepte sowie erste Empfehlungen für Informatikstandards diskutiert. Diese Entwicklungen sollten gute Voraussetzungen für eine dringend erforderliche informatische Bildung in allen Jahrgangsstufen bieten,

¹ Westfälische Wilhelms-Universität Münster, Didaktik der Informatik, Fliegenerstrasse 21, 48149 Münster, johanna.borsch@uni-muenster.de

² Anschrift s. o., marco.thomas@uni-muenster.de

³ Im Folgenden sprechen wir verkürzend von GI-Bildungsstandards, da sich diese Empfehlungen quasi als Standard etabliert haben und derzeit keine bundesweiten Standards für das Fach Informatik existieren.

⁴ Allerdings könnten diese aufgrund der geringen verfügbaren Stundenanzahl im gymnasialen Wahlpflichtbereich möglicherweise in einer reduzierteren Form verfasst werden als die bisherigen KLPs.

wobei einschränkend betont werden muss, dass Informatik bisher in NRW in keiner Jahrgangsstufe verpflichtend vorgesehen ist. Sofern Informatik als Wahl(pflicht)fach angeboten wird, steht es in starker Konkurrenz zu anderen Angeboten. Auf schulpolitischer Ebene wird die Notwendigkeit einer informatischen Bildung für alle Schülerinnen und Schüler⁵ offenbar weiterhin als gering eingeschätzt – auch in NRW. Es wird den Schulen überlassen, welche Angebote sie mit den verfügbaren Ressourcen durchführen. Gerade mit Blick auf die derzeitigen bundes- und landesweiten Anstrengungen im Rahmen einer eher medienpädagogisch ausgerichteten, so genannten »digitalen Bildung«, erscheint es uns erforderlich, die Situation der informatischen Bildung an den Schulen spezifischer zu beschreiben⁶. Diesem Anliegen dienen unsere Umfragen, die wir seit 2013 im Rahmen des Projekts KISS⁷ einmal im Jahr durchführen.

2 Ziele der Studie

Die Gegebenheiten (Personal, Schulprofil etc.) bezüglich einer informatischen Bildung an den Schulen der SI in NRW sind offenbar recht unterschiedlich. Es erscheint uns daher auch interessant, die Ausstattung an den Schulen für eine informatische Bildung zu explorieren. Vielfach ist bereits darauf hingewiesen worden, dass an den Schulen ein eklatanter Mangel an Informatikfachkräften existiert und sich diese Situation eher verschlechtern wird (s. z. B. [DM16], S. 17). In der aktuellen Studie ermitteln wir zudem die Zufriedenheit mit den Rahmenbedingungen für das Fach Informatik, da die Daten aus den qualitativen Fragen einer Lehrer- und Schulleiterumfrage im Jahr 2015 auf Kritik, insbesondere an den verfügbaren Ressourcen, hinweisen.

Gerade im Kontext einer Schulprofilierung sind die Sichtweisen von Entscheidungsträgern, die vermutlich eher selten universitär-informatisch geprägt sind, von Bedeutung. Dass es Aufklärungsbedarf zu den Elementen einer informatischen Bildung gibt, haben wir bereits mittels einer Studie auf dem Münsteraner Workshop 2016 aufgezeigt ([BTY16]). Erneut versuchen wir Aussagen zu den Angeboten für eine informatische Bildung zu gewinnen, wobei wir die Medienerziehung in die Untersuchung einbeziehen.

Unsere Studie untersucht die Bekanntheit und Akzeptanz der GI-Bildungsstandards bzw. der neuen Kernlehrpläne sowie den Stand einer Umsetzung an den Schulen. Die Realisierung der Kernlehrpläne steht und fällt mit dem Einsatz von Informatik-Fachkräften an den Schulen und mit den Möglichkeiten zur Fort- und Weiterbildung, sodass wir auch zu diesen Rahmenbedingungen in unserer Studie Fragen stellen. Zur Planung von gezielten Fortbildungsangeboten ist es hilfreich zu wissen, welche der eigenen Kompetenzen von den Lehrkräften als fortbildungsbedürftig angesehen werden.

⁵ Aus Gründen der Übersichtlichkeit verzichten wir im Folgenden auf die Nennung unterschiedlicher Gender, sofern der Sachverhalt dieses nicht erfordert.

⁶ Problematisch an den Schulstatistiken NRWs [NW16] ist beispielsweise, dass die Schulen weitestgehend entscheiden können, ob Angebote der Informatik zugerechnet werden oder nicht.

⁷ Kriterien zum Informatikunterricht von Schülerinnen und Schülern

Für Fortbildungen stehen an deutschen Schulen – unserer Ansicht nach – zu wenig Arbeitsstunden und Mittel zur Verfügung, obwohl Fortbildungen immanenter Bestandteil der schulischen Lehre sein sollten. Inwieweit sich E-Learning in Informatik-Lehrerfortbildungen integrieren lässt, wird mit dem verwendeten Fragebogen ebenfalls untersucht.

3 Untersuchungsmethodik

Die Erhebung der Daten erfolgt quantitativ mit einem standardisierten Fragebogen, der Online auszufüllen ist. Abhängig von den Eingaben werden relevante Frageblöcke ein- oder ausgeblendet. Als Druckfassung beträgt die durchschnittliche Länge des Fragebogens ca. sieben Seiten. Die geschätzte Bearbeitungszeit liegt bei ca. 20 Minuten.

Der Online-Fragebogen⁸ umfasst acht Blöcke zu den Themen „Allgemeines“, „Informatikangebot“, „Bewertung des Kernlehrplans Informatik“, „Bewertung der Bildungsstandards“, „Fortbildungsbedarf“, „Bereitschaft zu E-Learning“, „Anforderungen an Lehrerfortbildungen“ und „Zeiteinteilung für Fortbildungen“⁹.

Der erste Teil des Fragebogens fokussiert auf allgemeine Angaben zur Schule und zur Person der Teilnehmer. Informatiklehrern werden dabei vier zusätzliche Fragen zum beruflichen Werdegang und zur aktuellen Tätigkeit als Informatiklehrer gestellt. Der zweite Teil erfasst, wie zufrieden die Teilnehmer mit dem Angebot und der Ausstattung an ihrer Schule sind und ob es Angebote zur Medienerziehung oder informatischen Bildung in der SI gibt. Wird angegeben, dass es im Wahlpflichtbereich kein Angebot zur Informatik gibt oder wird zu dieser Frage „Sonstige“ angekreuzt, so wird jeweils eine zusätzliche Frage eingeblendet, die diese Angaben konkretisieren sollen. Teil 3 und 4 decken sich im Aufbau nahezu. Teilnehmer von Real-, Sekundar- und Gesamtschulen werden in Teil 3 zum Kernlehrplan befragt, während alle übrigen Schulformen, für die aktuell kein Kernlehrplan existiert, in Teil 4 in ähnlicher Weise zu den GI-Bildungsstandards befragt werden. Diese Unterscheidung findet auch im zweiten Teil von Block fünf statt, der erfasst, welchen Fortbildungsbedarf die Lehrer für sich in Bezug auf die Kernlehrpläne bzw. die GI-Bildungsstandards sehen. Durch dieses kontextabhängige Vorgehen wird auch der Gesamtumfang der Umfrage deutlich reduziert.

4 Datenerhebung

Gegenstand der Untersuchung sind Lehrer, die Informatik (im weitesten Sinne) unterrichten, und (stellvertretende) Schulleiter. Der Aufruf zur Umfrage wurde ab

⁸ Fragebogen und Auswertung sind online verfügbar unter <http://ddi.uni-muenster.de/ab/pu>

⁹ Die letzten drei Themen (Block 6-8) werden in einem gesonderten Beitrag unseres Arbeitsbereichs ausgewertet (s. a. K. Akao in diesem Band).

September 2016 zunächst über alle Bezirksregierungen NRWs gestartet und sollte an alle Haupt-, Real-, Gesamt- und Sekundarschulen sowie Gymnasien gehen. Im Dezember ist deutlich geworden, dass dieses Vorgehen dieses Mal nur teilweise unterstützt wurde, so dass wir einen erneuten Aufruf an diese Schulen via Faxversand gestartet haben. Die Umfrage ist Anfang Februar 2017 beendet worden.

Insgesamt haben wir 247 Rückmeldungen erhalten, aus denen wir die Teilnahme von 211 verschiedenen Schulen abgeleitet haben. Leider konnten wir auch mit dem aufwändigen Faxversand keine hohen Rückmeldequoten erzielen.

Schulform ¹⁰	H	R	S	Ge	Gym
Gesamt	456	559	114	314	625
Rückmeldungen (in %)	5	8	9	10	16

Tabelle 1: Anzahl Schulen bzgl. intendierten Population nach Schulformen. Quelle: [NW16]

Wie auch schon in den vergangenen Umfragen liegt die Rückmeldequote bei Gymnasien höher als bei den anderen Schulformen. Am niedrigsten ist die Rückmeldequote bei den Hauptschulen.

Schulform	H	R	S	Ge	Gym	Gesamt
Anzahl Informatiklehrer ¹¹	119	532	46	337	1383	2417
Rückmeldungen	11	41	7	22	117	202
in %	9	8	15	7	8	8

Tabelle 2: Rückmeldung von Lehrern für das Fach Informatik. Quelle: [NW16]

¹⁰ H = Hauptschule, R = Realschule; S = Sekundarschule, Ge = Gesamtschule, Gym = Gymnasium,

¹¹ Inkl. der Lehrer in der SII, da die amtlichen Schuldaten für das Schuljahr 2015/16 [NW16] keine diesbezügliche Unterscheidung machen.

Schulform	H	R	S	Ge	Gym	Sonstige	Gesamt
bin Lehrer für das Fach Informatik	11	41	7	22	117	4	202
unterrichte Informatik oder Fächer mit informatischen Inhalten	17	48	9	30	118	5	227 ¹²
unterrichte keine Fächer mit informatischen Inhalten	4	4	3	4	4	1	20
bin Schulleiter	3	9	3	3	5	1	24
bin stellvertretender Schulleiter	2	11	0	2	7	0	22

Tabelle 3: Teilnehmer der Umfrage nach Funktion(en) an der Schule

Gut 80% der Teilnehmer geben an, dass sie Lehrer für das Fach Informatik sind. 25 Lehrer unterrichten „Fächer mit informatischen Inhalten“, ohne dass sie angeben, Informatiklehrer zu sein. Des Weiteren können wir festhalten, dass 46 (stellvertretende) Schulleiter teilgenommen haben (14%), von denen 6 Schulleiter und 16 stellvertretende Schulleiter angeben, dass sie Informatiklehrer sind. Insgesamt gesehen passt unsere Stichprobe auf die intendierte Population.

5 Ergebnisse und Auswertung

Aufgrund der recht geringen Rückmeldequoten der Umfrage sind die erhaltenen Daten sehr vorsichtig zu interpretieren und keinesfalls als „repräsentativ“ anzusehen. Allerdings lassen sich durchaus interessante Trends erkennen und Fragen initiieren.

5.1 Schulausstattung

Während die amtliche Schulstatistik NRWs [NW16] zeigt, dass im Durchschnitt 1,2 Lehrer für das Fach Informatik pro Schule zur Verfügung stehen, ergibt sich in unserer Umfrage ein Durchschnitt von ca. drei Informatiklehrern pro Schule. In der Studie lag der Schnitt für jede Schulform über dem Schnitt aus der NRW-Statistik:

¹²Bei den informatiknahen Lehrern ist das Geschlechterverhältnis ca. 75 zu 25 für die Männer. Bei den Teilnehmern ohne informatische Fächer ist es eher andersherum: 40 zu 60 für die Frauen.

Schulform	H	R	S	Ge	Gym ¹³
Schulen gesamt	456	559	114	314	625
Informatiklehrer	119	532	46	337	1383
Informatiklehrer pro Schule nach amtlichen Schuldaten	0,3	1,0	0,4	1,1	2,2
Mittelwert in der Studie bzgl. Lehrer, die ein Fach mit informatischen Schwerpunkten unterrichten können (auch fachfremd)	2,6	2,6	2,4	2,9	2,8

Tabelle 4: Informatiklehrer an den Schulen. Quelle: [NW16]

Dies kann dahingehend interpretiert werden, dass vor allem informatikaffine Schulen an der Umfrage teilgenommen haben. Nur vier Teilnehmer geben an, dass sie keine Lehrer an ihrer Schule haben, die ein Fach mit informatischen Schwerpunkten unterrichten können.

Auch in unserer Umfrage wird deutlich, dass Informatiklehrer relativ spät in den Lehrerberuf einsteigen: Basierend auf dem Alter der Teilnehmer und der Zeitspanne der bisherigen Unterrichtstätigkeit für das Fach Informatik, ergibt sich die These, dass Informatiklehrer mit ca. 35 Jahren mit dem Unterrichtsfach starten. Zudem unterrichtet ein hoher Anteil an Lehrern fachfremd¹⁴ Informatik. Dies geht einher mit einem hohen Anteil von 34,2% an Nachqualifikationen (i. d. R. „Zertifikatskurse“ der Bezirksregierungen, die zu einer Unterrichtserlaubnis führen) im Fach Informatik, was direkte Auswirkungen auf den Unterricht und den Stellenwert des Fachs Informatik an den Schulen haben dürfte. Dieser Wert deckt sich in etwa mit dem Wert, den wir bereits 2015 mit einer Lehrerumfrage ermittelt haben. 2015 lag der Anteil bei 30,2 % (N=96).

5.2 Schulische Angebote

Um die Angebote zu einer informatischen Bildung an den Schulen präziser erfassen zu können, haben wir differenziert nach dem Angebot einer Medienerziehung, einer verbindlichen informatischen Grundbildung (VIG), einer fächerintegrierten informationstechnischen Grundbildung (FIG) und einem Informatikunterricht (IU) gefragt.

Überraschenderweise haben 35% der Schulen der Umfrageteilnehmer kein festes Stundenkontingent für eine Medienerziehung eingeplant. Die anderen Schulen bieten

¹³ In den Zahlen für die Gymnasien und Gesamtschulen sind in den amtlichen Schuldaten auch die SII-Lehrer enthalten.

¹⁴ Fachfremd bedeutet Unterrichten ohne Lehrbefähigung, d. h. ohne Staatsexamen in dem Fach. Dorothee Müller [DM16] hat in ihrer Dissertation lesenswert die Situation des Fachs Informatik in NRW dargestellt und Gründe für den Lehrermangel untersucht. Von schulpolitischer Seite wird der Lehrermangel im Fach Informatik als ein Argument gegen die Einrichtung eines Pflichtfachs Informatik in der Sekundarstufe I benannt (mündliche Überlieferung einer Stellungnahme der Schulministerin, 2016).

Medienerziehung zumeist über mehrere Stufen verteilt an, vornehmlich in den Stufen 5 bis 7. An Realschulen ist das Angebot zur Medienerziehung über die Schulstufen tendenziell schwächer ausgeprägt, während Sekundarschulen eher ein größeres Angebot zu haben scheinen. Eine verpflichtende informatische Grundbildung basierend auf einem schulinternen Konzept findet sich an immerhin 40% der Schulen, wiederum vorwiegend in den Stufen 5 bis 7. Das Angebot ist jedoch deutlich geringer als zur Medienerziehung. Es kann ein statistisch signifikanter Zusammenhang zwischen dem Angebot von Medienerziehung und verpflichtender informatischer Grundbildung in den unterschiedlichen Jahrgangsstufen festgestellt werden.¹⁵ Dies könnte darauf hindeuten, dass Schulen beide Bildungsziele in ein Pflichtfach integrieren oder der Begriff „Medienerziehung“ nicht klar von dem Begriff „verpflichtende informatische Grundbildung“ abgegrenzt wird.¹⁶ Ein ähnlich starker signifikanter Zusammenhang kann ebenfalls auch zwischen dem Angebot von VIG und verpflichtendem Informatikangebot festgestellt werden. Erstaunlich ist (erneut), dass 41% der Schulen der Ansicht sind, keine fächerintegrierte informationstechnische Grundbildung (FIG) anzubieten. Es lässt sich ebenfalls ein statistisch signifikanter Zusammenhang zwischen Medienerziehung und fächerintegrierter informationstechnischer Grundbildung erkennen.¹⁷

Erwartungsgemäß gibt es an nahezu 80% der Schulen kein Pflichtfach Informatik in der Sekundarstufe I. Im Wahlpflichtbereich der Sekundarstufe I wird Informatik an fast 50% der Schulen als „reiner“ Informatikunterricht angeboten. 18% der Schulen bieten Informatik im Wahlpflichtbereich nicht an und begründen dies zumeist mit dem Vorrang „traditioneller“ Fächer wie z. B. Französisch (49%) und dem Lehrermangel (es gibt bei uns keine Lehrer: 24%; nicht genug Lehrer für den Wahlpflichtbereich: 32%). Durchschnittlich werden 2-3 Stunden Informatik im Wahlpflichtbereich der Stufen 7-10 angeboten (die Stufen hängen von der jeweiligen Schulform ab).

In je einer Schulleiter- und einer Informatiklehrerumfrage 2015 haben wir die Frage gestellt: „Welche Verbesserungsvorschläge oder Anregungen fallen Ihnen zur Schulinformatik in NRW ein?“. Eine qualitative Analyse nach Mayring ergab die These, dass der geringe Anteil an verpflichtenden Informatikunterricht in der SI sowohl von Schulleitern als auch von Informatiklehrern kritisiert wurde. Ein Drittel der teilnehmenden Lehrer bzw. 50% der Lehrer, die die Frage beantwortet haben, wünschen sich eine verpflichtende SI-Informatik (Schulleiter: 13% der Teilnehmer; 32% der beantworteten Fragen). Ebenfalls kritisiert wurde in dieser Auswertung die Ausstattung der Schulen bzgl. Hardware und Software (Schulleiter: 11% / 28%; Lehrer: 13% / 20%). In der aktuellen Umfrage sind über 50% der befragten Lehrer mit dem Informatik-

¹⁵ Für die Jahrgangsstufen 5-8 ist der p-Wert für den Chi-Quadrat-Test und dem Test nach Fischer bei 0,000 und die erwartete Häufigkeit des Chi-Quadrat-Test in jeder Zelle größer als 5. In Stufe 9 und 10 ist die erwartete Häufigkeit in einer Zelle jeweils kleiner als 5 (25%). In Jahrgangsstufe 9/10 ist der p-Wert für den Chi-Quadrat-Test bei 0,007/0,005 und für den exakten Fischer Test bei 0,014/0,021. Man kann also immer noch von einem signifikanten Zusammenhang ausgehen.

¹⁶ Obwohl wir bei der Formulierung der Fragen sehr auf eine Unterscheidung in den Begrifflichkeiten geachtet haben.

¹⁷ Für Interessierte stellen wir die statistischen Kennzahlen mit der Auswertung der Daten wieder online auf unseren Webseiten bereit.

angebot an ihrer Schule, den curricularen Vorgaben (s. a. u.) und der technischen Ausstattung zufrieden¹⁸.

Darüber hinaus haben wir die Teilnehmer der oben genannten Umfragen gebeten, die informatischen Schwerpunkte ihrer Schule zur informatischen Bildung (Projektthemen, Wettbewerbsteilnahmen) kurz zu beschreiben. Häufig genannt wurden: AGs (z. B. zur Robotik), Informatikbiber, Bundeswettbewerb Informatik und Projektstage. Letztlich zeigt sich hier, dass die Lehrer bzw. die Schulen auf vielfältige Art und Weise Freiräume nutzen, um ihren Schülerinnen und Schülern eine informatische Bildung zu ermöglichen.

5.3 Kernlehrplan/Bildungsstandards

Ein Viertel der Teilnehmer von Real-, Gesamt- und Sekundarschulen geben an, dass sie die neuen Kernlehrpläne nicht gelesen haben (N=95). Nur 14% der teilnehmenden Schulen dieser Schulformen geben an, dass die Kernlehrpläne bereits in schulinterne Lehrpläne umgesetzt wurden. An ca. 55% der Schulen der Teilnehmer dieser Schulformen seien die seit 2015 gültigen Kernlehrpläne zur Informatik noch nicht umgesetzt worden. Die verbleibenden Teilnehmer geben an, dass er teilweise umgesetzt wurde. Man kann zwischen der geplanten Umsetzung und dem Stand der Umsetzung an den genannten Schulformen einen signifikanten Zusammenhang feststellen¹⁹. Eine These hierzu könnte sein, dass die Schulen, die den Kernlehrplan bisher noch nicht umgesetzt haben, kein informatisches Angebot im Wahlpflichtbereich haben und daher den Kernlehrplan auch nicht umsetzen. An den Gymnasien kennen nur 46% diese SI-Kernlehrpläne ganz oder teilweise.

Insgesamt lässt sich in der Studie die Tendenz erkennen, dass die Teilnehmer an Gesamt-, Real- und Sekundarschulen mit dem neuen Kernlehrplan, der an den Bildungsstandards orientiert ist, zufrieden sind. Sie stimmen den Aussagen „Der Kernlehrplan deckt meine inhaltlichen/kompetenzbezogenen Erwartungen“, „Der Kernlehrplan ist grundlegend für eine zeitgemäße Medienbildung“ und „Der Kernlehrplan lässt genügend Freiräume für Schwerpunktsetzungen“ tendenziell zu²⁰. Sie sind größtenteils (46% zu 40%) der Meinung, dass die Umsetzung in die Unterrichtspraxis relativ problemlos funktioniert. Darüber, ob es genügend Unterrichtsmaterialien gibt, sind sie sich eher uneinig (über 50% negieren diese Frage). Abschließend lässt sich eine starke Tendenz (65%) dahingehend feststellen, dass der Umfang an Fortbildungen bzgl. des neuen Kernlehrplans als nicht angemessen erachtet wird.

Die GI-Empfehlungen zu Bildungsstandards wurden von 66% der Befragten (teilweise)

¹⁸ Die Zufriedenheit ist sogar so groß, dass ganz unzufriedene Personen statistisch als Ausreißer erfasst werden. In der qualitativen Studie werden vermutlich eher die unzufriedenen Teilnehmer ihre Kritik bekundet haben.

¹⁹ Chi-Quadrat Test/ Test nach Fischer $p=0,000$. 24 der 26 Schulen die den Kernlehrplan bereits teilweise umgesetzt haben, planen dies auch weiterhin zu tun. Bei den Schulen, die den Kernlehrplan bisher nicht umgesetzt haben, planen ca. 50% die Umsetzung.

²⁰ Über 50% stimmen teilweise oder vollkommen zu.

gelesen (36% geben „Ja“ an; 30% „teilweise“). Für die Hauptschulen und Gymnasien existieren derzeit keine Kernlehrpläne für die SI-Informatik. An 23% dieser Schulen wurden die Bildungsstandards in einem schulinternen Lehrplan umgesetzt²¹, aber 53% der anderen 77% geben auch an, diese nicht umsetzen zu wollen. Es lässt sich darüber hinaus signifikant nachweisen, dass es einen Zusammenhang zwischen der Bekanntheit der Bildungsstandards an den Schulen und dem Willen zur Umsetzung gibt. Dabei deuten die statistischen Ergebnisse darauf hin, dass der Umsetzungswille von der Bekanntheit abhängt.²² Generell ist die Zustimmung zu den Bildungsstandards deutlich geringer als die der anderen Schulformen zu den Kernlehrplänen (vgl. o.).

5.4 Fortbildungen

Auffällig bei informatikaffinen Befragten ist vor allem der hohe Anteil (30%) von denen, die keine Fortbildungen gehabt haben. Im Mittel ergibt sich ein Schnitt von ca. drei Fortbildungstagen (Median 2 Tage) in den letzten 3 Jahren bei den Teilnehmern, die Informatik oder Fächer mit informatischen Inhalten unterrichten. Dies deckt sich auch mit den Angaben, wie oft Fortbildungen stattfinden sollten (1-2 Mal im Jahr)²³.

Generell scheint ein Bedarf an Fortbildungen in Bezug auf den neuen Kernlehrplan vorhanden zu sein. Dies geben immerhin fast 70% der Teilnehmer mit informatischen Fächern an Real-, Sekundar- und Gesamtschulen an. Zu den Bildungsstandards sind es 57% der Befragten. Zu allen Inhaltsbereichen des Kernlehrplans geben mehr als 50% der Teilnehmer mit „informatischen Fächern“ der Real-, Sekundar- und Gesamtschulen einen zumindest teilweisen Fortbildungsbedarf für sich an. Bezogen auf die Bildungsstandards (Gym, H, Sonstige) wird insbesondere für die Inhaltsbereiche „Algorithmen“ sowie „Sprachen und Automaten“ ein hoher Fortbildungsbedarf gesehen. Während für den Kernlehrplan (R, S, Ge) die Inhaltsbereiche eher gleichgewichtig hinsichtlich einer Fortbildung gesehen werden, variieren die Einschätzungen zu den Inhaltsfeldern der GI-Bildungsstandards stärker.

6 Fazit und Ausblick

Die GI-Bildungsstandards und deren Implementierung in Kernlehrpläne sind in NRW auf einem guten Weg und finden in der Lehrerschaft deutliche Akzeptanz, obwohl zu einigen Kompetenzerwartungen ein Fortbildungsbedarf seitens der Lehrkräfte signalisiert wird. Allerdings kann – aus unserer Sicht – nicht akzeptiert werden, dass

²¹ Die Frage zur Umsetzung der Bildungsstandards wurde nur Teilnehmern von Hauptschulen, sonstigen Schulen und Gymnasien gestellt. Die zur Kenntnis der Bildungsstandards wurde allen Teilnehmern der Umfrage gestellt.

²² Chi-Quadrat Test $p=0,005$. $\lambda = 0,250$ bei „Umsetzung“ abhängig (asymptotisch signifikant), $\lambda = 0,071$ bei „gelesen“ abhängig.

²³ Es zeigt sich eine Tendenz, dass der Wunsch nach Fortbildungstagen vergleichbar zur Realität durchgeführter Fortbildungstage ist (kein signifikanter Zusammenhang).

Schulen keine Angebote zur informatischen Bildung machen. Es fehlen für den Informatikunterricht in der Sekundarstufe I in NRW ausreichend Ressourcen in den Bereichen Fachpersonal, Fort- und Weiterbildung und zuletzt auch bei der entsprechenden Ausstattung von Schulen. Sicherlich sind diese Mängel kein Alleinstellungsmerkmal der Informatik, aber bei einem weniger etablierten Fach dürften sich diese gravierender als in anderen Schulfächern auswirken. Auch die Universitäten müssten sich stärker für Informatik-Lehramtsstudiengänge der Sekundarstufe I engagieren und nicht zuletzt – gemeinsam mit der KMK – Konzepte entwickeln, um mehr Studierende für Studiengänge zu gewinnen, die auf das Unterrichten von Informatik in der Schule vorbereiten. Es bleibt leider festzuhalten, dass das Erheben von Daten zu Forschungszwecken im Schulbereich durch schulpolitische Entscheidungen massiv erschwert wird. Eine Unterstützung von Umfragen durch die zuständigen Behörden könnte zu „repräsentativeren“ Ergebnissen führen, die für alle Beteiligten nützlich sind. Es wäre zu wünschen, dass in konstruktiven Gesprächen nach Lösungen gesucht wird, die für den Schulalltag verkraftbar sind. Die Tür zum Informatikunterricht sollte nicht verschlossen werden.

Literaturverzeichnis

- [BTY16] Borsch, Johanna; Thomas, Marco; Yomayuzza, Angélica (2016): Informatisches Bildung in den frühen Klassen der Sekundarstufe I. Eine explorative Studie in den Regierungsbezirken Münster und Detmold. In: Marco Thomas und Michael Weigend (Hg.): Informatik für Kinder. 7. Münsteraner Workshop zur Schulinformatik. 1. Aufl. Norderstedt: Books on Demand, S. 93–102.
- [DM16] Müller, Dorothea: Der Berufswahlprozess von Informatiklehrkräften. Dissertationsschrift, Wuppertal 2016. Online verfügbar unter <http://elpub.bib.uni-wuppertal.de/edocs/dokumente/fbc/mathematik/diss2016/mueller> (09.02.2017).
- [GI08] Gesellschaft für Informatik (Hg.) (2008): Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. Empfehlungen erarbeitet vom Arbeitskreis Bildungsstandards. Online verfügbar unter (20.03.2014) https://www.gi.de/fileadmin/redaktion/empfehlungen/Bildungsstandards_2008.pdf.
- [KR15] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen (2015): Kernlehrplan für die Realschule in Nordrhein-Westfalen - Wahlpflichtfach Informatik. Online unter <http://www.schulentwicklung.nrw.de/lehrplaene/lehrplannavigator-s-i/realschule/index.html> verfügbar (06.01.2017).
- [KG15] MSW NRW (2015): Kernlehrplan für die Gesamtschule/Sekundarschule in Nordrhein-Westfalen - Wahlpflichtfach Informatik. S. [KR15]
- [NW16] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen: Amtliche Schuldaten zum Schuljahr 2015/16. Online verfügbar unter (09.02.2017) <https://www.schulministerium.nrw.de/docs/bp/Ministerium/Service/Schulstatistik/Amtliche-Schuldaten/index.html>

Hochschule als außerschulischer Lernort für Schülerinnen und Schüler: Ein pragmatisches Angebot

Lars Prädel¹, Gerlinde Schreiber²

Abstract: Der vorliegende Beitrag stellt das Konzept und die Arbeiten im Informatik-Lernlabor vor, das Jugendliche unterschiedlicher Altersstufen anspricht und in verschiedensten Formaten (von 2h Schnuppern bis hin zum einjährigen Praktikum) einen praktischen Einstieg in informatikbezogene Alltagsthemen bietet. Exemplarisch wird die Konzeption und Durchführung von drei Angeboten ausführlicher ausgeführt: Ein Kompaktkurs im Rahmen einer Kinderferienakademie auf der Insel Wangerooge, ein mehrstündiges Schnupperangebot für Schulklassen sowie die Entwicklung von Facharbeiten der gymnasialen Oberstufe. Die Arbeiten im Lernlabor setzen auf frei verfügbaren Lernmodulen, frei verfügbarer Software sowie günstiger Hardware auf, die flexibel an unterschiedliche Bedürfnisse und Einsatzbedingungen angepasst werden können. Der Beitrag schließt mit den bisher gesammelten Erfahrungen zu den Erfolgsfaktoren dieser außerschulischen Angebote.

Keywords: Lernlabor; informatische Bildung; Bild der Informatik; ProgrammierEinstieg; persönliche Lernumgebung

1 Einleitung

Warum studieren Sie Informatik? Diese Frage wird den Erstsemestern jedes Jahr beim Studiumseingang gestellt. Die Antworten darauf sind vielfältig. Viele geben an, dass sie sich aus wirtschaftlichen Aspekten für das Studium entschlossen haben, einige geben die vage Aussage an, dass sie sich sehr viel mit Computern auseinandersetzen und mehr über die Funktionsweise lernen wollen. Wenige hingegen geben an, dass sie bereits Vorkenntnisse aus der Schule mitbringen oder Informatik aus dem familiären Bereich kennen und somit Berührungspunkte haben. Die Schlüsse, die daraus gezogen werden, sind einfach: Die Studierenden haben ein diffuses Bild, was in dem Studium auf sie zukommt und werden oftmals enttäuscht, wobei die hohen Abbruchquoten ein Beleg dafür sind. Auf der anderen Seite liegt der Verdacht nahe, dass es Personen gibt, die sich für ein Informatikstudium begeistern würden, wenn sie nur wüssten, was das bedeutet. Hier setzt das Informatik-Labor an. In unterschiedlichen Angeboten können Schülerinnen und Schüler mit oder ohne Lehrkräfte ebenso wie Studieninteressierte in die Informatik und insbesondere in die Programmierung hineinschnuppern.

¹ Hochschule Bremen, Flughafenallee 10, 28213 Bremen lars.praedel@hs-bremen.de

² Hochschule Bremen, Flughafenallee 10, 28213 Bremen gerlinde.schreiber@hs-bremen.de

Ziel des Lernlabors ist es, mehr Begeisterung für MINT-Fächer zu wecken, ihren Alltagsbezug deutlich zu machen und die Abbruchquote im Studium durch eine bessere vorhergehende Orientierung zu senken. Alle Angebote des Lernlabors orientieren sich an didaktischen Leitlinien, die in ähnlicher Form auch aus anderen Lernlaboren [In17, UF17] bekannt sind:

- Alle Themen sind praxisorientiert und haben einen klaren Anwendungshintergrund mit Bezug zum Umfeld der Jugendlichen.
- Die Jugendlichen arbeiten möglichst eigenständig in kleinen Teams.
- Die Arbeitsgruppen können selbst inhaltliche Schwerpunkte setzen, die ihren Interessen und ihrer Vorbildung angemessen sind.
- Die Betreuung erfolgt flexibel in Abhängigkeit von Bedarf und Nachfrage der Teams.

2 Umfeld und Ausstattung

Das Informatik-Lernlabor richtet sich an Schülerinnen und Schüler (mit oder ohne Lehrkräfte) ebenso wie an Studieninteressierte. Eine besondere Herausforderung stellen die unterschiedlichen Vorkenntnisse und die zeitliche Einschränkung dar. Die Teilnehmerinnen und Teilnehmer sollen mit einer neuen Aufgabenstellung konfrontiert werden, bei der zentrale Aspekte der Informatik, beziehungsweise der Programmierung, behandelt werden, die aber auch bei Interesse zu Hause weiter verfolgt werden können, ohne großen finanziellen Aufwand und teurem Equipment. Daher werden kostengünstige Mikrocomputer Raspberry Pi [Ra17] verwendet, die sich durch eine transportable Größe und vielseitige Möglichkeiten auszeichnen. So können diverse kostenlose Programme verwendet werden, die den Einstieg in die Programmierung erleichtern oder aber auch elektronische Schaltungen angeschlossen und mit einer Programmierschnittstelle verknüpft werden. Mikrocontroller, wie Arduino [Ar17], bieten auch einen Einstieg in elektronische Schaltungen und zentrale Aspekte der Programmierung.

Durch die Einbindung in die Fakultät für Elektrotechnik und Informatik besteht darüber hinaus die Möglichkeit, auf die im akademischen Lehrbetrieb genutzten Ressourcen zuzugreifen (beispielsweise Virtual Reality/Augmented Reality-Ausstattung, 3D-Scanner und 3D-Drucker). Als Ausblick auf das Studium können mit dem kostenfreien Programm Blender [Bl17] 3D-Animationen erstellt und bearbeitet werden.

3 Ausgewählte Angebote

Im folgenden sind einige exemplarische Angebote angegeben, die kontinuierlich durchgeführt werden.

3.1 Kompaktkurse

In einem Kompaktkurs wird eine Gruppe von Jugendlichen eine Woche lang mehrere Stunden pro Tag betreut. Ziel bei diesem Konzept ist, anhand einfacher Aufgaben einen Einstieg in die Programmierung zu erhalten und anschließend eigene Ideen der Teilnehmerinnen und Teilnehmer umzusetzen und sie dabei zu unterstützen. Dieses Konzept wird zum Beispiel in der Kinderferienakademie auf der Nordseeinsel Wangerooge umgesetzt. Dabei haben sowohl Jugendliche, die mit ihren Eltern Urlaub auf der Insel machen, als auch Jugendliche, die auf der Insel wohnen, die Möglichkeit daran teilzunehmen. In einer Gruppe von bis zu 15 Teilnehmerinnen und Teilnehmern im Alter von 12 - 16 Jahren werden in den Räumlichkeiten der Inselschule Wangerooge verschiedene kurze Projekte durchgeführt. Nicht nur aufgrund der Transportfähigkeit der Raspberry Pis und der Arduinos bieten sich Angebote mit diesen Geräten an. Nach einer Einführung in die Funktionsweise der Hardware und des Linuxsystems können unterschiedliche Projekte durchgeführt werden. Zum Beispiel kann ein Programm in der Programmiersprache Python [Py17] erstellt werden, welches die Möglichkeit unterstützt, die virtuelle Welt von dem Spiel Minecraft [Mi17a] zu verändern. In der Raspberry Pi Version [Mi17b] dieses Spieles ist es möglich, verschiedene Gebäude oder Landschaften aus unterschiedlichen Würfeln zu erstellen. Durch die Programmierschnittstelle zu Python können diese auch algorithmisch verändert werden. Da dieses Spiel bei Jugendlichen populär und somit das Spielprinzip von vornherein klar ist, wird ein einfacher Einstieg ermöglicht. Es entsteht der Eindruck, dass die Teilnehmerinnen und Teilnehmer ein neues Feature in dem Spiel erforschen, auf der anderen Seite aber mit ernsthafter Programmierung konfrontiert werden. Die Jugendlichen lernen einfache Konzepte der Programmierung und können die Ergebnisse direkt in dem Spiel verfolgen. So kann eine Schleife in Python dazu verwendet werden um eine Mauer zu erstellen. Auch die Verwendung von Variablen und anderen Kontrollstrukturen wird erlernt, indem die Position des Spielers abgefragt oder verändert wird. Nachdem kleinere Aufgaben gestellt werden, können die Teilnehmerinnen und Teilnehmer eigene Ideen verwirklichen und so eigene Spiele innerhalb der Minecraft-Welt erstellen.

3.2 Facharbeiten

In Zusammenarbeit mit der gymnasialen Oberstufe von Schulen werden jährlich Projektarbeiten vergeben. Diese Projektarbeiten werden je von zwei bis vier Schülerinnen und Schülern in dem Abiturjahrgang über einen Zeitraum von 2 - 3 Wochen bearbeitet. Bei diesen Facharbeiten bekommen die Teilnehmerinnen und Teilnehmer eine Aufgabe gestellt und bearbeiten diese nach einer kurzen Einführung selbstständig. Das Lernlabor fungiert als Kunde, der ein Projekt in Auftrag gibt, aber auch mithilft, dieses umzusetzen. Dabei werden die Schülerinnen und Schüler in den Hochschulkontext aufgenommen und können am Hochschulleben teilnehmen. Als Beispiel sei unter anderem ein ferngesteuertes Auto genannt, das zunächst aus Lego-Bauteilen konstruiert wird. Ein Arduino steuert zwei Motoren des Fahrzeugs an. Über eine Bluetooth-Verbindung wird eine Verbindung zu einem

Android-Smartphone erstellt und eine App, die in dem kostenfreien Programm App Inventor [MI17c] erstellt wird, steuert dieses Auto. Dabei werden vielseitige Konzepte vermittelt: die elektronische Schaltung zum Ansteuern der Motoren und des Bluetooth-Chips, das Programmieren von Microcontrollern inklusive Schleifen, Variablen und Kontrollstrukturen, das Erstellen einer Android-App innerhalb von App Inventor und die Herstellung einer Bluetooth-Verbindung. Die Teilnehmerinnen und Teilnehmer werden bewusst mit der Aufgabe alleine gelassen und müssen sich selbst das benötigte Wissen aneignen, haben aber kontinuierlich die Gelegenheit Fragen zu stellen. Der Vorteil ist, dass mit einfachen kostengünstigen Mitteln ein abgeschlossenes Projekt erstellt werden kann, das eindrucksvoll und vorzeigbar ist.

3.3 Schnupperangebot

Einen Tag die Hochschule kennenlernen und sehen, was Informatik bedeutet: Dieses Angebot richtet das Lernlabor an Schulklassen, aber auch an einzelne Schülerinnen und Schüler. Dabei wird eine Übersicht über die Möglichkeiten eines Studiums an der Hochschule vorgestellt, gefolgt von einer praktischen Arbeit. Damit wird ein Einblick in die Hochschule gewährt, aktuelle Projekte werden vorgestellt, aber es wird auch programmiert.

Dabei wird unter anderem eine Alarmanlage in Python programmiert, die ein Foto macht, sobald eine Lichtschranke passiert wird, und ein Alarmsignal ertönt. Dabei wird ein Raspberry Pi verwendet, bei dem eine elektronische Schaltung mit einem Fotowiderstand, einer LED und einem Summer angeschlossen ist. Mit einer entsprechenden Bibliothek von Python [Gp17] können diese Bauteile einzeln angesteuert werden. Der Wert von dem Lichtwiderstand wird in Variablen abgespeichert und abhängig von diesem Wert wird ein Alarm ausgelöst. So kann ein Blinken der LED in einer Schleife realisiert werden, die die LED ein- und ausschaltet, oder der Summer kann für eine bestimmte Zeit einen Ton erzeugen. Auch hier werden grundlegende Inhalte der Informatik und insbesondere des Programmierens in einer authentischen Entwicklungsumgebung einfach vermittelt.

Ein anderer Workshop widmet sich der 3D-Visualisierung. Dieser Workshop richtet sich eher an Schulklassen, die bereits einen Background in Informatik haben und einen Ausblick auf das Studium erlangen wollen. Dabei können auf die Ressourcen der Hochschule zugegriffen und 3D-Objekte in dem kostenfreien Programm Blender erstellt und animiert werden. Diese Objekte können dann mit einem 3D-Drucker ausgedruckt werden, oder es können Objekte und Personen mit dem 3D-Scanner erfasst werden. Auch hier liegt der Fokus darin, dass einfache Grundlagen der Computer-Graphik vermittelt werden, die aber auch zu Hause weiterverfolgt werden können.

4 Erfahrungen

„Im Informatikunterricht erhalten Schülerinnen und Schüler vielfältige Gelegenheiten zur Entwicklung und Ausbildung von Kompetenzen, die sie befähigen, ihr Leben in einer Informationsgesellschaft selbstbestimmt zu führen und zu gestalten“ [GI16]. Dieser Präambel der von der Gesellschaft für Informatik verabschiedeten Bildungsstandards ist unbedingt zuzustimmen - aber was ist mit der informatischen Bildung derjenigen Schülerinnen und Schüler, die auf ihrem schulischen Weg keinen Informatikunterricht durchlaufen? Hier setzen außerschulische Bildungsangebote wie das Vorgestellte an. Nach unseren Erfahrungen (Befragung am Ende der Angebote) sind folgende Aspekte der Angebotsgestaltung für die Jugendlichen besonders wichtig und bestätigen unsere didaktischen Leitlinien: Voraussetzung für die Teilnahme sind nicht konkrete Fachkenntnisse, sondern Interesse und Engagement. Die Themen sind unmittelbar verständlich und als praktische alltagsbezogene Aufgabenstellung einleuchtend. Die Betreuung im Lernlabor bietet Unterstützung beim Einstieg in das jeweilige Thema, begleitet danach kontinuierlich, geht auf individuelle Fragen ein und gibt Feedback zum Projektfortschritt. Die Betreuung drängt sich nicht auf, sondern schaltet sich auf Initiative der Jugendlichen ein. Sie wird als Unterstützung wahrgenommen und nicht als Kontrollinstanz. Aufgaben werden in selbst gewählten Teams bearbeitet. Zu beobachten ist, dass die Teilnehmerinnen sich bevorzugt zu reinen Mädchen-teams zusammenschließen. Im Team kann die inhaltliche Schwerpunktsetzung individuell in Abhängigkeit von Interessen und Vorkenntnissen festgelegt werden. Die Zuständigkeiten können sich im Verlauf verändern und weiterentwickeln. Im Lernlabor findet keine Leistungsbewertung nach dem Motto "wer ist wie weit gekommen" statt, sondern es gibt Anerkennung für Fortschritte und Wertschätzung für kreative eigene Ideen. Die Arbeiten finden an einem authentischen Lernort statt (innovative Hochschulumgebung mit Kontakten zu Lehrenden und Studierenden).

5 Fazit

Das Schülerlabor profitiert von der engen Kooperation mit einzelnen Schulen und Lehrkräften und von besonderen Rahmenbedingungen: Die Angebote werden von den Teilnehmerinnen und Teilnehmern freiwillig ausgewählt aus Interesse an Thema und Umgebung. Damit entfallen im Lernlabor die klassischen schulischen Motivationsprobleme. Die Angebote richten sich vor allem an kleinere Gruppen, deren eigene technische Ausstattung (nach dem Motto „Bring your own Device“) in die Aufgabenbeschreibung mit einbezogen werden kann. Die private Fortsetzung und Vertiefung von im Lernlabor durchgeführten Projekten wird dadurch erleichtert. Das Lernlabor besitzt nicht die Ressourcen zur Neuentwicklung von Angeboten oder der homogenen Ausstattung von Arbeitsgruppen mit aufwendiger Hard- oder Software. Stattdessen werden frei zugängliche Lernumgebungen und Aufgabensammlungen (wie [Re17], [Fr17]) gesichtet, ausgewählt und an die Vorkenntnisse und zeitlichen Rahmenbedingungen der jeweiligen Teilnehmerinnen und Teilnehmer angepasst. Damit

leistet das Lernlabor auf pragmatische Weise einen Beitrag, Schülerinnen und Schüler beim Aufbau einer persönlichen Lernumgebung zu unterstützen.

Literaturverzeichnis

- [Ar17] Arduino AG, <http://www.arduino.cc>, Stand: 14.02.2017.
- [Bl17] Blender Foundation, <http://www.blender.org>, Stand: 14.02.2017.
- [Fr17] Friends-of-Fritzing Foundation, <http://fritzing.org/projects/>, Stand: 14.02.2017.
- [GI16] GI Bildungsstandards Sek2, http://informatikstandards.de/docs/Bildungsstandards_SII.pdf, Stand: 14.02.2017.
- [Gp17] Gpiozero - Raspberry Pi Foundation, <http://gpiozero.readthedocs.io/>, Stand: 14.02.2017.
- [In17] Info-Sphere - Schülerlabor Informatik der RWTH Aachen, <http://schuelerlabor.informatik.rwth-aachen.de>, Stand: 14.02.2017.
- [Mi17a] Minecraft (Mojang Synergies AB), <http://minecraft.net>, Stand: 14.02.2017.
- [Mi17b] Minecraft: PI Edition (Mojang Synergies AB), <http://minecraft.net/en-us/edition/pi/>, Stand: 14.02.2017.
- [MI17c] MIT App Inventor - Massachusetts Institute of Technology, <http://appinventor.mit.edu/>, Stand: 14.02.2017.
- [Py17] Python Software Foundation, <http://www.python.org>, Stand: 14.02.2017.
- [Ra17] Raspberry Pi Foundation, <http://www.raspberrypi.org>, Stand: 14.02.2017.
- [Re17] Resources - Raspberry Pi Foundation, <http://www.raspberrypi.org/resources/>, Stand: 14.02.2017.
- [UF17] UFZ-Schülerlabor - Helmholtz Zentrum für Umweltforschung, <http://www.ufz.de/schuelerlabor/> Stand: 14.02.2017.

Aller Anfang ist schwer! Wie gelingt der Einstieg in den Informatikunterricht?

Dieter Engbring¹

Abstract: Dieser Aufsatz befasst sich mit dem Umstand, dass nicht allen SchülerInnen mit der erwünschten Geschwindigkeit und den erwarteten Erfolgen der Einstieg in den Informatikunterricht gelingt. Zu den Gründen werden in diesem Aufsatz Hypothesen vorgelegt, die sich aus Beobachtungen im Informatikunterricht sowie aus Gesprächen mit LehrerInnen und SchülerInnen ergeben. Zusätzlich werden alternative Vorgehensweisen vorgestellt, die nicht nur einen anderen Informatikanfangsunterricht beschreiben, sowie der Überprüfung der Hypothesen dienen.

Keywords: Anfangsunterricht, Programmierumgebungen und -sprachen, Objektorientierung

1 Einleitung

Nicht nur hinter vorgehaltener Hand, sondern ganz offen reden Informatik-LehrerInnen darüber, wie schwer es einigen ihrer SchülerInnen fällt, selbst die einfachen und grundlegenden Begriffe zu lernen und leichte Aufgaben zu lösen. In der fachdidaktischen Literatur gibt es bislang keine Untersuchungen, die den Gründen und dem Ausmaß dieser Befunde nachgehen. Auch dieser Aufsatz kann hierzu allenfalls erste Hypothesen liefern, aus denen dann eine systematische Untersuchung entwickelt werden soll (2). Diese Hypothesen werden durch einen Alternativvorschlag für den Einstieg (3) und einem Forschungsansatz zur Überprüfung der Hypothesen und zur Evaluation des alternativen Zugangs ergänzt (4). In einem Fazit mit Ausblick wird auf die jetzt anstehenden Forschungsarbeiten hingewiesen (5).

2 Befunde zum Informatikunterricht in der Einführungsphase

Das Implementieren und damit auch die Programmiersprache sowie die Entwicklungsumgebungen spielen eine große Rolle in der alltäglichen Praxis des Informatikunterrichts in der Einführungsphase der gymnasialen Oberstufe, die immer noch für viele SchülerInnen der Informatik-Anfangsunterricht ist. Zwar ist es das weiterreichende Ziel, die dahinterstehenden Techniken und Denkweisen der informatischen Modellierung zu vermitteln, die auch – siehe Computational Thinking [Wi06] oder ähnliche Ansätze – über die Informatik hinaus anwendbar sind. Ansätze, diese Kompetenzen ohne Implementierung, frei von einer Programmiersprache und einer dedizierten ggf. auch einer für unterrichtliche Zwecke gestaltete (Entwicklungs-)Umgebung zu erreichen, greifen allerdings wohl

¹ Institut für Informatik, Universität Bonn, Römerstr. 164, 53117 Bonn

zu kurz. Dafür gibt es eine Reihe von lernpsychologischen und fachlichen Gründen, auf die hier nicht gesondert eingegangen werden kann. Denn es ist das Anliegen dieses Aufsatzes, die alltägliche Praxis des Informatikunterrichts, die das Implementieren im erheblichen Umfang beinhaltet, zu analysieren. Denn Beobachtungen der alltäglichen Praxis haben ergeben, dass dem Implementieren sehr viel Zeit eingeräumt wird, ohne dass die Lernenden dabei zusätzliche Kompetenzen erwerben. Mehr noch: In der Phase des Implementierens wird die ohnehin schon erhebliche Heterogenität größer, weil insbesondere viel Zeit dafür ver(sch)wendet wird, syntaktisch korrekten Code zu erzeugen, und dabei die nicht intrinsisch motivierten SchülerInnen sehr vorsichtig und langsam agieren. Darüber hinaus befassen diese sich außerhalb des Unterrichts kaum bis gar nicht mit den Inhalten. Diese Unterschiede scheinen durch den nun vollzogenen Wechsel zur Objektorientierung größer geworden zu sein.

Im Folgenden wird die dahinter stehende komplexe Gemengelage in (untersuchbare) Teilaspekte zerlegt. Dies geschieht, wie in der Einleitung schon betont, in Form von Hypothesen, die anhand von Symptomen oder Befunden aus dem Informatikunterricht hergeleitet werden. Dazu wird zunächst dem Aspekt der Objektorientierung nachgegangen, von dem man sich einen einfacheren Zugang versprochen hatte, weil sie „der natürlichen Denkweise am nächsten kommt“ [Sc95, S. 186] und der möglichst dem imperativen Paradigma vorangehen sollte [Di07]. In diesem Zusammenhang hat es in den letzten Jahren eine ganze Reihe von Untersuchungen und Publikationen gegeben, die den Nutzen der Objektorientierung in ihrer Ambivalenz diskutieren (2.1). Danach wird auf populäre Entwicklungsumgebungen für den Informatikunterricht eingegangen, die u. a. das Ziel haben, den vielleicht zunächst nicht so motivierten SchülerInnen den Zugang zum Programmieren einfacher bzw. überhaupt schmackhaft zu machen. Es zeigt sich, dass diese Herausforderungen mit Technik (allein) nicht zu lösen sind (2.2). Zusätzlich wird auf grundsätzliche Schwierigkeiten des Informatik-Anfangsunterrichts hingewiesen, die sich daraus ergeben, dass man systematisch in einige Bausteine algorithmischer Sprache einführt (2.3). Ein Zwischenfazit beendet diesen Abschnitt (2.4).

2.1 Objektorientierung im Anfangsunterricht

Im Jahr 2009 erschien ein Aufsatz von Kortenkamp, Modrow, Oldenburg, Poloczek und Rabel [KM09], in dem sie sich kritisch mit der Objektorientierung im Informatikunterricht auseinandersetzen. Er endet mit diesem Satz:

„Die geringe Zahl von SchülerInnen, die Informatik in der Qualifikationsphase belegt, ist möglicherweise nicht nur den Zwängen des Zentralabiturs geschuldet, sondern auch eine Reaktion darauf, dass der Unterricht (immer noch) nicht an den Bedürfnissen der SchülerInnen orientiert ist.“ [KM09, S. 47]

Sie argumentieren in dem Aufsatz, dass OO nicht nur an den Bedürfnissen der SchülerInnen vorbeigeht, sondern auch an denen der Fachwissenschaftler, da auch die Vorbereitung auf ein Hochschulstudium nur unzureichend geleistet wird. Letzteres ist ein Aspekt, auf den Börstler schon zwei Jahre zuvor hingewiesen hatte [Bö07]. Außerdem schreiben

Kortenkamp und anderen:

„Die Behandlung von OOM ohne OOP steht vor einem Problem, das dem der Mengenlehre in der „Neuen Mathematik“ in den Siebziger Jahren des vorigen Jahrhunderts ähnelt: Die SchülerInnen lernen neue Wörter zur Beschreibung von Dingen, die sie schon kennen; sie erschließen sich damit aber keinen neuen Möglichkeiten, etwas zu tun oder zu verstehen.“ [KM09, S. 43]

Diese Analogie zur Mengenlehre erscheint insofern statthaft, da es vielen SchülerInnen in der Tat nicht gelingt, die Informatikbegriffe Klasse, Unterklasse, Objekt, Attribute und Attributwerte zu definieren und sinnvoll in ihre unterrichtlichen Äußerungen einzubauen. Dies scheint auch Folge des Zugangs; dies ist zugleich Inhalt von *Hypothese 1*: Es wird versucht, ohne Anschauung an einem Programm nur durch Bezug auf Fahrzeuge oder Tiere diese Begriffe (insbesondere unter Einbeziehung der Vererbung) vorab zu klären. Dass eine Klasse Teil des Programm-Codes ist, der diesen strukturiert und dass die Objekte (vielleicht sollte man besser von *Instanzen* sprechen) zur Laufzeit erzeugt werden, wird dabei zu wenig deutlich hervorgehoben. Insbesondere wird kaum darauf eingegangen, wie der Programm-Code im Computer verarbeitet wird.

Viele, die in der Objektorientierung eine Hoffnung auf einen Informatikunterricht gesehen haben, der besser zugänglich ist, beziehen sich auf einen Aufsatz von Schwill, der 1995 Programmierstile im Anfangsunterricht analysiert hat. Er schlussfolgert:

„Wir favorisieren für die Einführung in die Informatik den objektorientierten Ansatz vor allem aus drei Gründen: Erstens erfüllt dieses Paradigma unbestrittenmaßen informatisch orientierte Forderungen nach zeitgemäßem Unterricht mit mächtigen Konzepten ... Zweitens ist dieser Ansatz im Sinne des didaktischen Prinzips der Fortsetzbarkeit, ein zentrales Merkmal eines nach dem Spiralprinzip organisierten Curriculums, auf höheren Niveau beliebig ausbaufähig. ... Drittens – dies scheint aus pädagogischer Sicht der wichtigste Pluspunkt – ordnet sich der objektorientierte Stil in besonderer Weise harmonisch den elementaren kognitiven Prozessen unter, die beim Denken, Erkennen und Problemlösen im menschlichen Gehirn ablaufen.“ [Sc95, S. 183]

Letztlich beruht seine dritte Schlussfolgerung darauf, dass er ein kognitionspsychologisches Experiment zur Problemlösung (von Duncker aus den 1930er Jahren) einer Deutung zuführt, die Aspekte der Objektorientierung enthält. Einer dieser Aspekte, den Duncker *funktionale Bindung* nennt, behindert jedoch den Problemlöseprozess. Diese muss durch eine alternative Sicht auf die Objekte, durch eine andere *funktionale Bindung*, im Problemlöseprozess überwunden werden. Damit erscheint die Schlussfolgerung weit weniger zwingend, da objektorientiertes Denken den Widerspruch, dass die *funktionale Bindung* den Problemlöseprozess ggf. behindert, eigentlich nicht auflösen kann.

Die Vorteile der Objektorientierung ergeben sich, den Beobachtungen im Informatikunterricht folgend, vielmehr aus einem anderen Umstand. Dies ist dann *Hypothese 2*: Den SchülerInnen fällt es leichter die Objekte und deren Aufgaben in einem Anwendungskontext zu benennen und zu beschreiben als von Beginn an Algorithmen zu beschreiben. Damit wird die Zerlegung der Aufgabenstellung 'einfacher', als wenn sie direkt algorithmisch z. B. durch *stepwise refinement* erfolgen würde. Um dieses zu erreichen, müssen die Beispiele jedoch sinnvoll gewählt werden ('Hello World' oder mathematische

Aufgaben sind dann wohl kein sinnvoller Einstieg).

Zugleich muss man sich eingestehen, dass Algorithmen und imperative Programmierung nicht obsolet sind. Die Einführung der Objektorientierung hat dazu geführt, dass die SchülerInnen ein Mehr an Begriffen und Konzepten lernen müssen. Denn Objektorientierung ist – so wie sie in den Schulen am Beispiel der Sprache Java unterrichtet wird – vor allem eine *imperative Programmierung*++.² Aus dieser Erkenntnis ergibt sich ein pragmatischerer Zugang wie er 2008 von Hu vorgeschlagen wurde. Er überschreibt seinen Aufsatz mit „Just say 'a class defines a Data Type'“ [Hu08]. Objektorientierung wird in diesem Sinne eine andere, vielleicht sogar elegantere bzw. eine größere Vielfalt zulassende Methode, Datentypen zu strukturieren und miteinander in Beziehung zu setzen.

Insofern ergibt sich *Hypothese 3*: Die Vorteile von OOM und OOP werden wohl insbesondere dann erkennbar, wenn man tatsächlich a) Vererbung benötigt wird, da durch eine Klassenhierarchie wie z. B. im Bereich der Komponenten für GUIs ein erheblicher Rationalisierungseffekt eintritt, b) die Algorithmen als Methoden sehr viel enger und auch in kompakterer Schreibweise (die Punktschreibweise) an die Datenobjekte gekoppelt werden können sowie c) in komplexeren Kontexten, an denen eine Vielzahl³ von Objekten beteiligt ist und dadurch das Problem/die Aufgabe zerlegt werden kann.

Damit ist die ideologisch geführte Diskussion um „Objects later“ bzw. „Objects first“ für die Praxis weniger interessant. Vor allem die Untersuchungen von Ehler (zum Teil mit Schulte) zeigen, dass letztlich keine Präferenzen angegeben werden können (vgl. hierzu die Zusammenfassung in [Eh12, S. 184ff]) Man kann zwar beobachten, dass Objektorientierung einen anderen Zugang zur Problemlösung erfordert als einen rein imperativen. Letztlich werden auch bei der Nutzung objektorientierter Sprachen Algorithmen entworfen. Dies verweist dann auf die *Hypothese 4*: In der imperativen Modellierung und Programmierung sozialisierte Personen lassen sich deswegen weniger auf objektorientiertes Modellieren ein, weil sie darum wissen, dass am Ende doch noch algorithmisiert werden muss. Diese durchaus schwierige Arbeit wollen sie so schnell wie möglich erledigen. Außerdem vermuten sie in der Suche nach Objekten und der Zuordnung von Aufgaben einen Overhead, der bei kleinen und bereits sehr gut mathematisierten bzw. algorithmisierten Kontexten, die oftmals für den Einstieg genutzt werden, auch vorhanden ist.

2.2 Programmierumgebungen für Anfänger

In den letzten 30 Jahren sind immer wieder neue Programmierumgebungen für den Informatik(anfangs)unterricht vorgestellt worden. Oftmals werden den SchülerInnen grafische Objekte zur Verfügung gestellt, die vorprogrammierte Funktionen enthalten, die dann weiter genutzt werden sollen. Anhand von zwei Beispielen wird im Folgenden kurz erläutert, dass hierdurch die Lernprozesse zum einen in Bezug auf die Mensch-Maschine-Interaktion erleichtert werden sollen und zum anderen Motivation geschaffen

² Vgl. C und C++. M. a. W.: Der Umfang des Lernstoffes wurde erhöht.

³ Für Anfänger erscheinen drei unterschiedliche Objekte eine Vielzahl, die aber zugleich erschließbar ist.

werden soll, sich überhaupt mit dem Programmieren zu befassen. *Hypothese 5*: Die Vielzahl solcher Umgebungen scheint ein Indiz für den Bedarf zu sein und für die weiterhin vorhandene Unzufriedenheit mit den bisherigen Umgebungen. Hier gibt es Prozesse, die man technisch unterstützt haben möchte, ggf. sind diese Wünsche auf didaktischen Kontext bezogen. Die Anforderungen sind im Anfangsunterricht andere als später. Daher wird immer wieder versucht eine „eierlegende Wollmilchsau“ zu erstellen.

Die beiden Beispiele, die im Folgenden näher betrachtet werden, sind *Alice*, das kaum eine Rolle (mehr) spielt und zum anderen *Greenfoot*, das derzeit auch unterstützt durch ein in 2.3 zu thematisierendes Schulbuch gerade eine große Rolle spielt.

Zu Alice: Pausch ist sozusagen der Vater von Alice. Da man darum wusste, dass er nicht mehr lange zu leben hat, wurde ihm die Gelegenheit gegeben, eine *Last Lecture* zu halten, in der er u. a. auf Alice einging. Sein Ziel war es, die Lernenden zu überlisten.

„So Alice is a project that we worked on for a long, long time. It's a novel way to teach computer programming. Kids make movies and games. The head fake, again, we're back to the head fakes. The best way to teach somebody something is to have them think they're learning something else. ... the head fake here is that they're learning to program but they just think they're making movies and video games.“⁴

Ein solcher *head fake* kann nur ein kurzfristig wirksamer Lernanreiz sein. Da Alice zudem nur vordergründig objektorientiert ist, sollte man sich an ihr nicht weiter orientieren (vgl. hierzu auch [DM09, En11]). In Alice (ähnlich wie z. B. in *Scratch*) wird das *Coden* durch Bau- bzw. Puzzlesteine unterstützt, sodass man schneller zu syntaktisch korrektem Code kommt. Dies schafft zwar ein wenig Raum für Kreativität, die jedoch mit den typischen Vorgehensweisen im Informatikunterricht im Konflikt steht (siehe 2.3).

Greenfoot ist hingegen eine Plattform, die in vielen Kursen (erfolgreich) genutzt wird. Dieser Aufsatz auf *BlueJ*, der auch von Kölling und Co entwickelt wurde, setzt letztlich den Ansatz von Mini-Welten fort, wie er durch *Logo-Turtle*, durch *Karol, the robot*, den Käfer *Kara* und viele mehr propagiert wird. In *Greenfoot* sind viele Implementationen in den Basisklassen *World* und *Actor* versteckt, wodurch die dahinterstehende Komplexität, mit der Anfänger auch nicht konfrontiert werden sollten, auch nicht erkennbar ist. Diese miteinzubeziehen würde wohl auch dazu führen, dass die SchülerInnen demotiviert werden. Die Implementation der Basisklassen der Szenarien (Roboter, Rover, Spinnen etc.) sind jedoch nicht versteckt, was im unterrichtlichen Geschehen u. a. dazu führen kann, dass gerade von pfiffigeren SchülerInnen die Szenarien manipuliert werden und korrekte Aufgabenlösungen nicht funktionieren. Zudem muss man *Greenfoot* irgendwann verlassen und auf eine andere Umgebung wechseln. Hier zeigt sich, wie sehr die gewonnenen Kenntnisse an die Umgebung gekoppelt sind und in andere Umgebungen, selbst wenn man nur zu *BlueJ* wechselt, nicht übertragen werden können (vgl. hierzu [En13]).

Hypothese 6: Der Versuch, mit Technik Motivation zu stiften, hat seine Grenzen. Letztlich werden die SchülerInnen mit nicht sonderlich kognitiv anspruchsvollen Aufgaben

⁴ Vgl. z.B. <http://theoncologist.alphamedpress.org/content/12/11/1374?trendmd-shared=1>

(komplexerer) objektorientierter und algorithmischer Modellierungen und Implementierungen konfrontiert, deren direkter Sinn sich nicht jedem erschließt. Der Sinn dieser Vorgehensweise liegt wohl insbesondere darin, in die Elemente einer Programmiersprache einzuführen, was sich auch im Folgenden zeigt.

2.3 Typische Vorgehensweisen im Anfangsunterricht

Greenfoot wird derzeit von vielen benutzt, da es anders als seine Basis *BlueJ* Anfänger besser unterstützt. Für NRW sind zur Nutzung von *Greenfoot* Schulbücher (z. B. [KL14] für die Einführungsphase) erschienen, die von vielen genutzt werden. Hier kann und soll keine ins Detail gehende kritische Bestandsaufnahme dieses Werkes erfolgen. Es soll nur kurz darauf hingewiesen werden, dass das Vorgehen in diesem Buch, wie zuvor auch im Einführungsband von Schriek [Sc05] („Stifte und Mäuse“, ähnlich wie das in NRW aktuell diskutierte *GLOOP*, ein grafischer Aufsatz auf *BlueJ*) oder auch anderen Büchern, die Reihenfolge der Sachlogik der Programmiersprache folgt. Die Inhalte Blockstrukturen, Schleifen, Verzweigungen, Variablen, Datenstrukturen werden in eine sachlogisch korrekte Reihenfolge gebracht. *Hypothese 7*: Im Informatikunterricht steht weniger das Erlernen bzw. Einüben des algorithmischen Problemlösens oder von informatischem Denken, vielmehr das Erlernen einer Programmiersprache im Vordergrund. Die SchülerInnen lösen nicht wirklich Probleme, sondern nur Aufgaben, die in ihrer Reihenfolge der Logik des Erlernens der Programmiersprache untergeordnet sind. Die Zusammenhänge zum Problemlösen werden nicht erkannt. Denning, Tedre und Youngpradit haben erst kürzlich in einem 'Viewpoint' in der CACM darauf hingewiesen, dass dieser Transfer bisher lediglich behauptet, aber nie nachgewiesen wurde. [DT17]

Außerdem müssen die SchülerInnen konstruierend vorgehen. Dies hat sicher lernpsychologische Vorteile, hat aber mit Blick auf die zu verwendenden Instrumente (die Entwicklungsumgebungen) auch Nachteile, da nicht zu hundert Prozent syntaktisch korrekte Lösungen zurückgewiesen werden. Aber selbst wenn das nicht geschieht, kommt es auch weiterhin zu Fehlern zur Laufzeit (*Stack overflow*, *Null pointer exception* ...) bzw. zu unerwünschten Systemverhalten, deren Korrektur nicht nur ein profundes Verständnis der internen Datenverarbeitungsprozesse erfordern, das (s. o.) nicht unbedingt zuvor entwickelt wurde. *Hypothese 8*: Dieser Wechsel zwischen den Ebenen ist eine Abstraktionsfähigkeiten abnötigende kognitive Leistung, die sich wiederum in einem Kontext stattfindet, der nur dem Erlernen einer Programmiersprache und weniger dem Erwerb von Problemlösekompetenzen dient. Die SchülerInnen werden lediglich gefordert, aber nicht notwendigerweise gefördert.

2.4 Zwischenfazit

Programmieren Lernen ist nicht nur der heimliche, sondern der offensichtliche Lehrplan zu Beginn der Einführungsphase. Darüber hinaus gehende Lernziele werden kaum erreicht. Dieser Lehrplan soll durch didaktische Tricks oder bunte Animationen schmack-

haft oder besser zugänglich gemacht werden. Diese Effekte sind jedoch nicht nachhaltig, auch weil es *nicht* das Bedürfnis der allermeisten SchülerInnen zu sein scheint, Programmieren lernen zu wollen. Dass sie dabei auch konstruktiv tätig werden müssen, aber von dem Instrument, mit dem sie arbeiten, immer wieder zurecht oder gar zurückgewiesen werden, trägt nicht zur Motivation bei; im Gegenteil! Erfolgserlebnisse und Sinnstiftung fehlen gleichermaßen. Zumindest eine solche Sinnstiftung wird durch den im Folgenden darzustellenden Zugang versucht.

3 Ein alternativer Zugang

Vergleicht man diese Schwierigkeiten des Informatikunterrichts mit anderen Fächern, ist der konstruktive Zugang ungewöhnlich. In vielen anderen Unterrichtsfächern steht eher die Analyse von Gegenständen, Phänomenen und Situationen der Natur, der Kultur oder der Zivilisation im Vordergrund. Im naturwissenschaftlichen Unterricht setzt man dabei im besten Fall auf entdeckendes Lernen in Form von Experimenten. Auch im Musikunterricht wird eher analysiert denn komponiert, ggf. wird musiziert, aber auch das wohl seltener, da man hier auf Probleme der Vorbildung und der starken Differenzierung stößt (Begabung).⁵ Aber selbst, wenn man im Fach Musik 'konstruktiv' vorgeht; die Instrumente würden den Versuch nicht zurückweisen. Allenfalls sind Disharmonien vernehmbar. Im Kunst- und – so es ihn gibt – im Technikunterricht wird konstruktiv gearbeitet, allerdings auch dort ohne grundsätzliche Zurückweisungen durch die Instrumente. Also eher rekonstruktiv bzw. analytisch vorzugehen, ist eine Facette dieses Zugangs.

Die andere Idee ist, die Aufgaben in ein Software-Projekt einzubetten. Das ist nicht neu und orientiert sich grob an den Vorstellungen von Magenheim [Ma08]. Zugleich wird der Versuch unternommen, diesen hochkomplexen und bislang nicht wirklich operationalisierten Ansatz für den Anfangsunterricht zu adaptieren. Verschiedene Versionen bzw. Entwicklungsstufen einer Software werden zyklisch analysiert und dann weiterentwickelt. Es wird mit der Analyse eines sehr einfachen und kurzen Programm-Codes gestartet. Es werden mithin zuerst die Bestandteile im Programm-Code aufgespürt und mit den korrekten Begriffen benannt, ähnlich wie man es Fach Deutsch mit Prosa, Lyrik aber auch Sachtexten macht, wobei dort der Vokabelapparat ein viel größerer ist. Erst nach Klärung der Begriffe, Konzepte und der Verarbeitungsmechanismen wird dann selbst implementiert.

Als Kontext für das Software-Projekt dient ein Brettspiel. Die Befragung von Rabel und Oldenburg [RO09] verweist darauf, dass dies durchaus an die Bedürfnisse der SchülerInnen anschließt, obschon die sich sicher eine hippere Spieleentwicklung wünschen würden.⁶ Am Beispiel des Spiels Kniffel (Knobeln, Yathzee) wird im Folgenden dieser Ansatz konkretisiert, wobei Kniffel nur ein Beispiel unter vielen möglichen ist. Hier erscheint Zugänglichkeit für die SchülerInnen (einfache Regeln, wenige aber unter-

⁵ ... auch eine Analogie zur Informatik?

⁶ . Auch Diethelm konkretisiert ihre Vorstellungen an einem Gesellschaftsspiel [Di07].

scheidbare Objekte, die zum Teil baugleich sind) jedoch in besonderer Weise gegeben.

Man startet zunächst mit der Implementation eines Würfels, der als Klasse (als Programm-Code und als UML-Klassendiagramm) den SchülerInnen gegeben wird. Code wie auch Diagramm werden auf ihre Bestandteile hin analysiert. Diese für jede Klasse gleiche Struktur wird ausführlich erarbeitet und die Beziehungen zwischen Code und Klassendiagramm können entdeckt werden. Der Code wird instanziiert und damit kann der Würfel auch „gerollt“ werden. Da die Klasse eine Grafikkomponente enthält (hiermit werden also auch Ideen der Mini-Welten aufgegriffen) sehen die SchülerInnen, dass etwas passiert und auch was. An dieser Stelle sind Experimente mit dem Code (nicht nur Sechser-Würfel, andere GUI ...) erwünscht. In dieser Phase sollte man sich Zeit lassen, die Beziehungen von Code und Programm mit den dazugehörigen Begrifflichkeiten zu klären, ohne dass man den SchülerInnen auch noch ein konstruktives Vorgehen abverlangt. Die Begriffe Objekt, Klasse etc. werden in ihren informatischen Kontext gestellt.

In einem zweiten Schritt soll der Übergang zum Würfelbecher stattfinden, der auf der Grundlage der Erkenntnisse der ersten Phasen von den SchülerInnen selbst modelliert und dann implementiert werden soll. An dieser Stelle können verschiedene Varianten der 'Kooperation' der Objekte Becher und Würfel diskutiert werden. Die SchülerInnen werden erkennen, dass die Klasse Würfel erweitert werden muss, um den wechselseitigen Bezug von Würfel und Becher zu implementieren. Am Ende dieser zweiten Phase, wenn die SchülerInnen den Würfelbecher implementiert haben, der dann nur die Würfel rollt, die auch im Würfelbecher sind, können sie bereits viele der z. B. in dem Lehrplan NRW genannten Kompetenzen erworben haben, da sie Teil des Unterrichts waren.⁷

Es wäre dann zu überprüfen, inwieweit diese Kompetenzen entwickelt wurden. Außerdem wäre zu untersuchen, wie SchülerInnen mit Vorkenntnissen mit diesem Zugang umgehen und wie motiviertere und begabtere SchülerInnen mit der Intensität des zunächst analytischen Vorgehens umgehen. Das Szenario scheint jedoch genügend viele Möglichkeiten zur Binnendifferenzierung zu bieten, bei der sich die 'Besseren' z. B. schon der Auswertung von Würfelergebnissen (ist es eine *Straße*, *Full House* ...) widmen oder einen digitalen Notizzettel erstellen (der alle Einträge zulässt, wie sein Vorbild aus Papier oder nur korrekte bzw. mit dem Würfelergebnis konsistente). Als besondere Herausforderung könnte ein „Computerspieler“ implementiert werden.

4 Alltagspraxis: Ein Forschungsansatz zur Prozessbeobachtung

Es wurde schon mehrfach betont, dass es hier nur Hypothesen (begründet oder plausibel) formuliert wurden und dass auch der gerade vorgestellte alternative Ansatz noch der Evaluation bedarf. Evaluation bedeutet – wie gerade ausgeführt – Kompetenzmessung, aber nicht nur. In dem hier vorzustellenden Ansatz der Alltagspraxis geht es um die

⁷ Diese können hier aus Platzgründen nicht aufgezählt werden. Vgl. http://www.schulentwicklung.nrw.de/lehrplaene/upload/klp_SII/if/KLP_GOST_Informatik.pdf S. 21-23

Erforschung der (alltäglichen) Prozesse im Unterricht, für die viele der Hypothesen noch genauer zu Forschungsfragen zu verdichten wären. Dazu gehört die Beobachtung der Aktivität der SchülerInnen (der Unterrichtsprozesse) als wichtige zu ergänzende Komponente. Da einige der Tätigkeiten an bzw. mit den Rechnern erfolgen, könnten via Eye-Tracking oder der Aufzeichnung von Tastatur und Mausevents Daten erhoben werden, die zwar keine absolute Bedeutung haben, aber möglicherweise durch begleitende qualitative Forschung mit relativen Bedeutungen versehen werden können. Zu diesem Ansatz gehören auch freie Beobachtungen, in denen man versucht weitere Phänomene aufzuspüren und dann in anderen Beobachtungen mit anderen Personen wiederzuentdecken (ethnografische Studien). Begleitet werden sollte das Ganze durch Interviews mit LehrerInnen sowie SchülerInnen, die dann qualitativ analysiert werden und mit deren Ergebnissen die Hypothesen und Forschungsfragen weiter präzisiert werden.

5 Fazit mit Ausblick

In diesem Aufsatz wurden auf der Grundlage von einer ganzen Reihe von Beobachtungen zum Informatikunterricht – auch im Vergleich zu anderen Fächern – Hypothesen zu den besonderen didaktischen Herausforderungen des Informatikunterrichts aufgestellt. Dazu wurde ein Ansatz vorgestellt, der nun evaluiert werden kann. Eine möglichst große Anzahl von LehrerInnen sowie SchülerInnen sollten damit konfrontiert werden. Deren Umsetzungen sollten mit den Prozessen und Erträgen konventionellen Informatikunterrichts verglichen werden, sofern sie vergleichbar sind. Alle anderen Informatik-DidaktikerInnen werden hiermit aufgefordert, sich an diesen Studien zu beteiligen.

Insgesamt wird damit ein empirischer Forschungsansatz vorgelegt, der dazu dienen soll, die durchaus widersprüchlichen Phänomene (sehr motivierte und ganz wenig motivierte SchülerInnen, große sowie ausbleibende Lernerfolge ...) zu systematisieren und über alternative Vorgehensweisen nachzudenken. Auch weitere neuralgische Stellen, wie z. B. die Behandlung von Algorithmen mit (dynamischen) Datenstrukturen erfordern einen eher analytisches und weniger konstruktives (= modellierendes und implementierendes) Vorgehen. Der Platz in diesem Beitrag reicht dafür nicht aus, dieses darzustellen.

6 Literatur

- [Bö07] Böstler, J.: Objektorientiertes Programmieren – Machen wir irgendwas falsch? In: Sigrid Schubert (Hrsg.): Didaktik der Informatik in Theorie und Praxis. INFOS 2007, 12. GI-Fachtagung Informatik und Schule GI-Edition - Lecture Notes in Informatics (LNI), P-112. Bonner Köllen Verlag (2007)
- [Di07] Diethelm, I: "Strictly models and objects first" - Unterrichtskonzept und -methodik für objektorientierte Modellierung im Informatikunterricht, Dissertation, Universität Kassel, Fachbereich Elektrotechnik/Informatik, 2007 <http://kobra.bibliothek.uni-kassel.de/bitstream/urn:nbn:de:hebis:34-2007101119340/1/DissIraDruckfassungA5.1.pdf>

- [DM09] Dohmen, M., Magenheim, J., Engbring, D.: Kreativer Einstieg in die Programmierung - Alice im Informatik-Anfangsunterricht. In: Peters, I. (Hrsg.): Informatische Bildung in Theorie und Praxis, Beiträge zur INFOS 2009, 13. GI-Fachtagung - Informatik und Schule, S.69-80, Berlin (LOG IN Verlag) 2009
- [DT17] Denning, P.J.; Tedre, M.; Youngpradit, P.: The Profession of IT. Misconception about Computer Science. Communications of the ACM. March 2017/Vol. 60, No. 3, S. 31 – 33. doi:10.1145/3041047
- [Eh12] Ehlert, A.: Empirische Studie: Unterschiede im Lernerfolg und Unterschiede im subjektiven Erleben des Unterrichts von Schülerinnen und Schülern im Informatik-Anfangsunterricht (11. Klasse Berufliches Gymnasium) in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First und OOP-Later), Dissertation, FU Berlin, 2012 http://www.diss.fu-berlin.de/diss/receive/FUDISS_thesis_000000035764
- [En11] Engbring, D.: Untersuchungen und Bewertungen zum Einsatz von Alice im Informatikunterricht. In: M. Weigend, M. Thomas, F. Otte (Hrsg.): Informatik mit Kopf, Herz und Hand. Praxisbeiträge zur INFOS 2011. ZfL-Verlag. Münster, S. 81 - 90
- [En13] Erst nehmen wir Greenfoot. Und dann BlueJ? In: Breier, N., Stechert, P., Wilke, T. (Hg.): INFOS 2013. 15. GI-Fachtagung Informatik und Schule. Praxisband. Kiel Computer Science Series. 2013/3. S. 29 – 39
- [Hu08] Hu, C.: Just say 'A class defines a Data Type'. In: Communications of the ACM, Vol. 51 No. 3, Pages 19-21 doi: 10.1145/1325555.1325560
- [KL14] Kempe, T.; Löhr, A.: Informatik 1. Schoeningh-Verlag. Paderborn. 2014
- [KM09] Kortenkamp, U; Modrow, E.; Oldenburg, R.; Poloczek, J; Rabel, M.: Objektorientierte Modellierung - aber wann und wie? Zur Bedeutung der OOM im Informatikunterricht. LOG IN 160/161, 2009. S. 41-47
- [Ma08] Magenheim, J.: Systemorientierte Didaktik der Informatik Sozio-technische Informatiksysteme als Unterrichtsgegenstand? In Kortenkamp; U.; Weigand; H.G.; Weth, T. (Hrsg.): Informatische Ideen im Mathematikunterricht, Franzbecker Hildesheim 2008. S. 17 - 36
- [RO09] Rabel, M; Oldenburg, R.:Konzepte, Modelle und Projekte im Informatikunterricht – Bewertungen und Erwartungen von Schülern und Studenten. In: Bernhard Koerber (Hrsg.): Zukunft braucht Herkunft. 25 Jahre »INFOS – Informatik und Schule« INFOS 2009, 13. GI-Fachtagung »Informatik und Schule« GI-Edition - Lecture Notes in Informatics (LNI), P-156. Bonner Köllen Verlag (2009), S. 146 -155
- [Sc95] Schwill, A.: Programmierstile im Anfangsunterricht. In: S. Schubert (Hrsg.): Innovative Konzepte für die Ausbildung 6. GI-Fachtagung Informatik und Schule. Springer. Berlin Heidelberg. S. 178-187
- [Sc05] Schriek, B.: Informatik mit Java: Eine Einführung mit BlueJ und der Bibliothek Stifte und Mäuse. Band 1. Nili-Verlag. Werl
- [Wi06] Wing, J.: Computational Thinking. Communications of the ACM. March 2006/ Vol. 49, No. 3, 33-35

Fünf Argumente für einen grafischen ProgrammierEinstieg - eine Studie über vier Jahrgangsstufen

Nadine Bergner¹, Thiemo Leonhardt¹ und Ulrik Schroeder¹

Abstract: Grafische Programmierwerkzeuge sind ein beliebter Weg mit Kindern und Jugendlichen in die Programmierung einzusteigen. Ein grafischer Einstieg soll die Schülerinnen und Schüler altersgerecht ansprechen und ihnen aufgrund der Vermeidung von Syntaxfehlern erleichtern, eigene Projekte schnell und kreativ umzusetzen und dadurch motivierend wirken. In diesem Artikel werden die vermuteten Chancen und auch mögliche Gefahren eines grafischen ProgrammierEinstiegs wissenschaftlich untersucht. Dazu werden fünf Argumente beleuchtet: Begeisterungsfähigkeit, Vermittlung eines interessanten Bildes der Disziplin, Erwartungen der ProgrammierEinsteiger, Einfluss auf die wahrgenommene Schwierigkeit der Disziplin und Vermittlung eines kreativen Berufsbildes “Informatiker-in”. Um diese Argumente wissenschaftlich zu untersuchen, wurden in 2015 und 2016 über 700 Kinder und Jugendliche vor und nach einem Einstiegsworkshop mit den Werkzeugen “Scratch” und “App Inventor” mittels Online-Fragebögen befragt. Die Ergebnisse decken auf, welche Auswirkungen die Workshops auf das Bild der Informatik sowie die Vorstellungen über Informatikerinnen und Informatiker haben.

Keywords: Grafische Programmierung, quantitative Evaluation, Scratch, App Inventor, Fragebogenstudie, Schülerlabor, Bild der Informatik, Berufsbild.

1 Motivation

Der Einstieg in die Programmierung kann inner- und außerhalb der Schule sowie in unterschiedlichen Altersstufen stattfinden. Den unterschiedlichen Gegebenheiten gemein ist, dass ein Weg gefunden werden muss, Kindern und Jugendlichen die Grundlagen der Programmierung zielgruppengerecht zu vermitteln.

Ein auch in Deutschland weit verbreiteter Ansatz ist die grafische Programmiersprache bzw. -umgebung *Scratch*. Scratch lässt den Lernenden Freiheiten mittels grafischer Programmierbausteine eigene interaktive Geschichten oder Spiele zu entwickeln, in denen Objekte bewegt werden können, Musik abgespielt und sogar mit dem Nutzer interagiert werden kann. Besonders durch das umfangreiche Angebot an ansprechenden und kindgerechten Grafiken werden Kinder der Klassenstufen 4 bis 8 mittels Scratch motiviert ihre eigenen Ideen in kleinen Programmen selbstständig zu verwirklichen. Das Werkzeug Scratch ist didaktisch sehr gut reduziert, kann durch das kindliche Design allerdings für ältere Jugendliche demotivierend wirken (vgl. [Ma10], [Ut10]).

¹RWTH Aachen, Informatik 9 - Lerntechnologien und Fachdidaktik Informatik, 52074 Aachen, {bergner, leonhardt, schroeder}@informatik.rwth-aachen.de

Eine weniger spielerische Möglichkeit des grafischen ProgrammierEinstiegs stellt der *App Inventor* des MIT bereit. Hier werden keine Katzen in bunten Unterwasserwelten, sondern Android-Apps für Smartphones und Tablets programmiert. Durch den Alltagsbezug der mobilen Geräte können Kinder und Jugendliche in einem sehr breiten Altersspektrum angesprochen und motiviert werden (vgl. [Wol1]).

Insbesondere durch den Bedeutungsgewinn der Programmierung für die Allgemeinbildung der Kinder könnte sich der Einsatz dieser Werkzeuge zukünftig verstärken. Offen bleibt dabei die Frage nach den Effekten solch grafischer ProgrammierEinstiege, welche Untersuchungsgegenstand dieses Artikels sind.

2 Forschungsfragen & -design

Beim Einsatz von Lernwerkzeugen stellt sich immer die Frage des Effekts sowie des Erfolgs bei der Zielgruppe, also den Lernenden. Dazu werden in diesem Paper die folgenden fünf Argumente, welche sich aus der oben genannten Literatur wie aus den Erfahrungen im Schülerlabor InfoSphere ableiten lassen, mittels einer Fragebogenstudie zu Einstiegsworkshops mit den Werkzeugen “Scratch” und “App Inventor” diskutiert:

1. Ein grafischer ProgrammierEinstieg kann die Lernenden für die Informatik begeistern.
2. Grafische Programmierung vermittelt das Bild einer interessanten und spannenden Wissenschaft Informatik.
3. Grafische Programmierung trifft die Erwartungen der ProgrammierEinsteiger.
4. Der grafische ProgrammierEinstieg hat Einfluss auf die wahrgenommene Schwierigkeit der Disziplin.
5. Grafische Programmierung vermittelt ein kreatives Berufsbild “Informatiker-in”.

Um Kinder und Jugendliche an die Informatik und speziell den Bereich der Programmierung heranzuführen, werden im Informatik-Schülerlabor InfoSphere an der RWTH Aachen seit 2010 über 30 verschiedene (in der Regel eintägige) Workshops angeboten, an denen Schulklassen und Kurse kostenfrei teilnehmen. Das Themenangebot beinhaltet Module für den Erstkontakt in der Grundschule und Unterstufe wie auch Module zur Vertiefung bzw. Anknüpfung an den bestehenden Informatikunterricht in der Mittel- und Oberstufe. Die Themen umfassen Facetten und Aspekte der theoretischen (z. B. zu Kryptographie-Verfahren), technischen (z. B. Schaltungen mit Mikrocontrollern) und praktischen (z. B. Erwerb von Programmierkompetenzen in unterschiedlichen Altersstufen) Informatik. Damit das punktuell angestoßene Interesse weiter ausgebaut werden kann, werden Kindern und Jugendlichen privat (an Wochenenden, in den Ferien) Möglichkeiten der tiefergehenden Einarbeitung geboten.

In dieser Auswertung bilden drei Module zum grafischen ProgrammierEinstieg für die Zielgruppe der Programmieranfänger der Klassenstufen 6 bis 9 den Fokus.

Das Forschungsdesign zur Evaluation der obigen fünf Argumente besteht aus einer quantitativen Forschungsmethodik mittels Online-Fragebögen², wobei der Pretest etwa ein bis zwei Wochen vor dem Besuch des Workshops in der Schule ausgefüllt wird und der Posttest direkt im Anschluss erhoben wird. Im Folgenden wird erläutert, mit welchen Fragen die fünf Aspekte beleuchtet werden.

Argument 1 - Begeisterungsfähigkeit

In dem Online-Fragebogen werden die Teilnehmerinnen und Teilnehmer explizit gefragt, wie ihnen der Einstieg in die (grafische) Programmierung gefallen hat und ob sie wieder an einem solchen Workshop teilnehmen möchten. Dazu steht den Lernenden eine fünfstufige Skala von „trifft gar nicht zu“ bis „trifft voll zu“ zur Beantwortung zur Verfügung. Analog wird ihre Zustimmung zur Aussage „Ich werde mich in Zukunft verstärkt mit Themen der Informatik beschäftigen.“ abgefragt.

Argument 2 - Vermittlung eines interessanten Bildes der Disziplin

Die Kinder und Jugendlichen werden vor und nach dem Besuch im InfoSphere dazu befragt, welches Bild der Disziplin Informatik sie haben. Dies geschieht zum einen, indem sie im Freitext drei Begriffe zur Informatik notieren. Zum anderen halten sie ihre Einstellung zu verschiedenen vorgegebenen Aussagen über die Informatik fest, indem sie einen Schieberegler zwischen zwei Gegenpositionen platzieren. Die hier relevanten Gegenpole sind „Informatik ist langweilig versus spannend.“, „Informatik ist interessant versus uninteressant.“ und „Informatik ist eintönig versus abwechslungsreich.“.

Argument 3 - Passung zu den Erwartungen der ProgrammierEinsteiger

Hier werden die Schülerinnen und Schüler auf einer fünfstufigen Skala danach befragt, wie exakt der Workshop ihren Erwartungen entsprochen hat. Darüber hinaus werden die Freitextangaben zu den Erwartungen im Pretest und die Verbesserungsvorschläge zum Workshop und die Themenwünsche für die Zukunft im Posttest ausgewertet.

Argument 4 - Einfluss auf die wahrgenommene Schwierigkeit der Disziplin

Für dieses Argument werden Angaben vor und nach dem Workshop verglichen. Relevant ist die Einschätzung zur Aussage „Informatik ist schwierig versus einfach.“.

Argument 5 - Vermittlung eines kreativen Berufsbildes “Informatiker-in”

Nach derselben Methodik wird die Vorstellung über Informatikerinnen bzw. Informatiker vor und nach dem Workshop erhoben. So können die Befragten den Schieberegler zur Aussage „Informatiker-innen müssen kreativ sein.“ nahezu stufenlos von „stimme gar nicht zu“ bis „stimme voll zu“ platzieren. Gleiches gilt für die Aussage „Informatiker-innen kreieren viel Neues.“.

² Weitere Details zum Forschungsdesign und dem konkreten Fragebogen können eingesehen werden unter [Be15].

3 Umsetzung der Workshops

Um eine wissenschaftliche Reflexion der Antworten der Teilnehmerinnen und Teilnehmer zu ermöglichen, werden in diesem Abschnitt die beiden verwendeten Werkzeuge „Scratch“ und „App Inventor“ wie auch die Ausgestaltung der Workshops im Schülerlabor InfoSphere beleuchtet.

3.1 Vergleich der Tools Scratch & App Inventor

Beide Werkzeuge zum grafischen ProgrammierEinstieg werden kostenlos, browserbasiert angeboten. Eine wichtige Gemeinsamkeit für diesen Forschungsansatz sind die optisch ähnlichen Programmierblöcke (siehe Abb. 1 für den Vergleich grundlegender Programmierbausteine), wobei speziell für jüngere Lernende die Option der breiten Sprachwahl (inklusive Deutsch) im Werkzeug „Scratch“ einen Vorteil gegenüber dem „App Inventor“ darstellt, welcher nicht in Deutsch verfügbar ist.



Abb. 1: Vergleich der grafischen Programmierbausteine (links Scratch, rechts App Inventor)

Im Gegensatz zu Scratch, welches 2007 speziell für Kinder entwickelt wurde, sollte der App Inventor ursprünglich Informatikkurse an Universitäten unterstützen und startete dort 2009 seine Pilotphase. Ein weiterer Unterschied ist, dass bei Scratch Objekte in einer Mikrowelt programmiert werden, welche direkt neben der Programmierfläche sichtbar ist, so dass die Lernenden stets in der gleichen Ansicht arbeiten. Im App Inventor gibt es dagegen zwei getrennte Ansichten: Designer und Blocks. Der Designer dient der Gestaltung des User Interface und der Auswahl aller Komponenten (Buttons, Grafiken, Töne, Timer, GPS). Darauf aufbauend stehen im Blocks Editor allgemeine und komponentenspezifische Blöcke (abhängig von den eingefügten Elementen im Designer) zur Programmierung der Funktionalität zur Verfügung. Eine Besonderheit bei der Arbeit mit dem App Inventor ist das Live-Testing, welches über einen integrierten Emulator oder auch direkt auf einem Android-Smartphone bzw. -Tablet möglich ist. Die von den Lernenden entwickelten Apps können somit in die Realität (auf das Smartphone) überführt werden und sind nicht auf eine künstliche Mikrowelt im Browser beschränkt. Dieser Unterschied muss bei der späteren Auswertung insbesondere hinsichtlich des Bildes der Disziplin berücksichtigt werden.

3.2 Beschreibung der Schülerlabor-Module

Nachdem die beiden Werkzeuge beschrieben und verglichen sind, werden hier die konkreten Workshops im Rahmen des InfoSphere vorgestellt, anhand derer die Effekte grafischer Programmierwerkzeuge evaluiert wurden.

Das Modul zu Scratch nennt sich „*Spielend Programmieren lernen mit Scratch*“³ und wird für Klassen der Unter- und Mittelstufe angeboten. Das didaktische Konzept hinter diesem Workshop ist ein Gruppenpuzzle. Dies bedeutet, dass jeweils vier Kinder ein Team bilden, dessen Ziel es ist, am Ende des Tages gemeinsam ein Spiel zu programmieren. Noch vor dem Start wird allen Teilnehmerinnen und Teilnehmern gemeinsam die Webseite Scratch präsentiert und gezeigt, wie die Bausteine zusammengesetzt werden, ohne vertieft auf die Bedeutung einzelner Befehle einzugehen. In der ersten Phase des Gruppenpuzzles teilen sich die Teams auf, sodass jedes Teammitglied gemeinsam mit Mitgliedern anderer Teams Experte auf einem bestimmten Gebiet wird. Dabei entstehen Expertengruppen für die Themenbereiche „Bewegung und Zeichnen“, „Fühlen und Variablen“, „Grafik und Aussehen“ und „Klang und Kommunikation“. In diesen Expertenteams lernen sie mit Hilfe kleinschrittiger Anleitungen und anhand konkreter Beispiele eine Auswahl der Befehle kennen. In der zweiten Phase des Gruppenpuzzles entwickeln die vier Teammitglieder gemeinsam ein Spiel, zu dem die grobe Spielidee und mögliche Erweiterungsansätze gegeben sind. Dabei trägt jedes Teammitglied eine Verantwortung, da jede Expertin bzw. jeder Experte für ein bestimmtes Thema zuständig ist, welches für die Umsetzung des Spiels benötigt wird. In dieser Phase können die Kinder sehr kreativ arbeiten, da Scratch zahlreiche grafische und auch akustische Optionen zur Ausgestaltung des Spiels bietet. Zum Abschluss präsentieren alle Teams ihre Spiele vor der Gruppe und können diese anschließend gegenseitig austesten. Darüber hinaus ist es möglich, dass die Teilnehmerinnen und Teilnehmer ihre Spiele zur weiteren Bearbeitung mitnehmen.

Zum App Inventor gibt es im Schülerlabor InfoSphere zwei unterschiedliche Module. Das Modul „*Erste App programmieren*“⁴ ist als Einstieg für Klassen der Stufen 6 bis 8 konzipiert, wohingegen das Modul „*InfoSphere goes Android*“⁵ als Fortsetzung oder für Anfänger der Stufen 8 bis 10 angeboten wird. Der Unterschied liegt im Detailgrad der Anleitungen wie auch in der Übersetzung aller englischen Begriffe. Die Sozialform in beiden Modulen ist die Partnerarbeit an einem Laptop und mit einem Smartphone. Auch hier starten beide Module mit einer kurzen Präsentation des Werkzeugs.

Im Modul „*Erste App programmieren*“ wird in drei Schritten eine Zeichen-App mit Foto-Funktion entwickelt. Mittels des Arbeitsblattes „Zeichnen für Anfänger“ wird programmiert, wie ein Foto aufgenommen und als Hintergrund gespeichert werden kann. Im Blatt „Zeichnen für Fortgeschrittene“ wird die App um Buttons erweitert, so dass die Stiftfarbe aus den vier Grundfarben gewählt werden kann und auch bereits Punkte und Linien in unterschiedlichen Strichstärken gezeichnet werden können. In dem optionalen Zusatzblatt „Zeichnen für Profis“ wird die App um eine komplette Farbpalette erweitert.

Das Modul „*InfoSphere goes Android*“ hingegen startet nach der Einführung mit einer ersten Phase, in der alle Zweiertteams eine einfache „Hallo InfoSphere“-App entwickeln,

³ Modulbeschreibung „Scratch“: <http://schuelerlabor.informatik.rwth-aachen.de/module/scratch>

⁴ Modulbeschreibung „Erste App“: <http://schuelerlabor.informatik.rwth-aachen.de/module/erste-app>

⁵ Modulbeschreibung „InfoSphere goes Android“: <http://schuelerlabor.informatik.rwth-aachen.de/module/is-goes-android>

die kleinschrittig angeleitet wird und den Umgang mit dem App Inventor übt. Anschließend hat jedes Team die Wahl zwischen den vier Spielideen „AngryBlob“, „Asteroids“, „FlappyBird“ und „MoleMash“, welche alle an existierende Apps angelehnt und daher den Schülerinnen und Schülern unter Umständen bereits bekannt sind. Hierbei wird der Schwierigkeitsgrad der Apps auf den Anleitungen angegeben, so dass die Teams nach der „Hallo InfoSphere“-App diesen selbst auswählen können. Die Anleitungen sind hier weniger kleinschrittig, so dass den Lernenden mehr Verantwortung über den eigenen Lernprozess übertragen wird. Auch haben die Lernenden die Möglichkeit ihre App nach eigenen Ideen weiterzuentwickeln.

4 Auswertung der Evaluation

In den Jahren 2015 und 2016 wurden für die vorgestellten drei Module zum grafischen ProgrammierEinstieg 694 Pre- und 720 Posttestdatensätze erhoben. Da die Vorbefragung in der Regel im Vorfeld in der Schule ausgefüllt wird, kommt es vor, dass Kinder (z. B. aufgrund von Krankheit) nicht an beiden Erhebungszeitpunkten einen Fragebogen ausfüllen. Aus dieser Problematik heraus ergibt sich, dass nur 564 Pre-Post-Datensätze korrekt zugeordnet werden konnten. Da zur Beforschung der fünf Argumente Informatikeinsteiger im Fokus stehen, wurde die Stichprobe für die Auswertung auf Schülerinnen und Schüler der Klassenstufen 6 bis 9 eingeschränkt. Dies ergibt einen Stichprobenumfang von N = 471 (siehe auch Abb. 2).

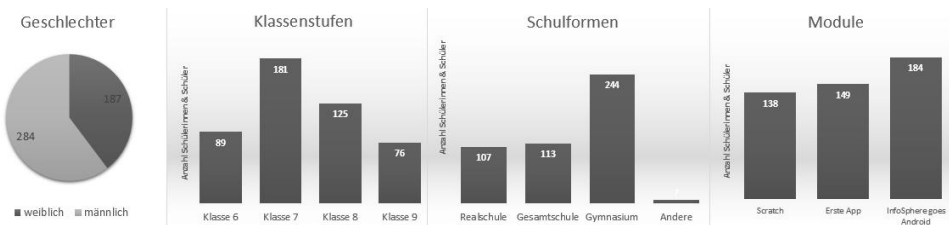


Abb. 2: Stichprobenbeschreibung

Argument 1 - Begeisterungsfähigkeit

Im Schnitt werden alle drei Module von den Teilnehmerinnen und Teilnehmern positiv bewertet (siehe Abb. 3 a)). Die Antworten hängen dabei signifikant vom Geschlecht der Kinder ($p < .05$), jedoch nicht vom Alter, der Klassenstufe und dem besuchten Modul ab. Die Daten der Mädchen zeigen mit einem Mittelwert von $M=2,44$ ($s=1,380$), dass den Mädchen das Modul signifikant besser gefallen hat als den Jungen mit $M=2,53$ ($s=1,598$)⁶. Obwohl die Differenzen nicht signifikant sind, zeigt ein Vergleich der Mittelwerte, dass das Modul „Scratch“ am beliebtesten und „InfoSphere goes Android“ am unbeliebtesten ist, was möglicherweise mit dem unterschiedlichen

⁶ Die Kodierung geschieht mittels „trifft voll zu“ (1) bis „trifft gar nicht zu“ (5).

Schwierigkeitsgrad der Module oder der tendenziell älteren Zielgruppe zusammenhängen könnte.

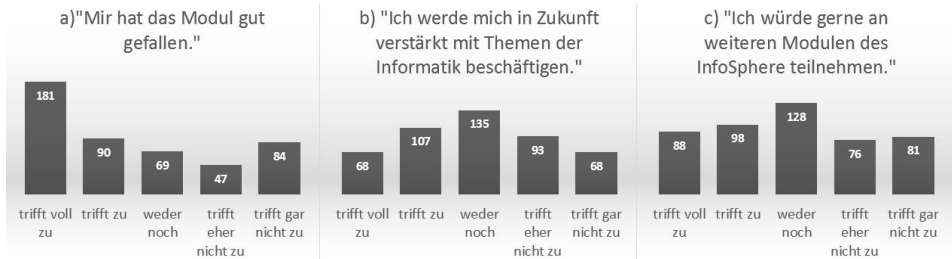


Abb. 3 Auswertung bezüglich Begeisterungsfähigkeit

Bezüglich der Aussage, ob sie sich in Zukunft verstärkt mit Themen der Informatik beschäftigen möchten, ergibt sich die Verteilung aus Abb. 3 b). Auch die Antwort auf diese Frage ist wieder signifikant vom Geschlecht abhängig ($p < .01$). Entgegen der obigen Aussage wollen sich hierbei die Jungen mit $M=2,87$ ($s=1,326$) eher zukünftig mit der Informatik beschäftigen als die Mädchen mit $M=3,12$ ($s=1,135$). Hier wird deutlich wie stark der, im Vorfeld generierte, Wunsch der Kinder bereits ausgeprägt ist und wie begrenzt dabei eine punktuelle Maßnahme wirkt. Darüber hinaus ist die Bewertung dieser Aussage signifikant abhängig von der Klassenstufe ($p < .05$). Schülerinnen und Schüler niedriger Klassenstufen geben deutlicher an, sich verstärkt mit Informatik beschäftigen zu wollen. Dies könnte in den geringeren Vorerfahrungen der jungen Zielgruppe begründet sein, so dass diese erst durch das Modul einen Einblick in die Informatik erhalten und durch ihre Erfolgserlebnisse bestärkt wurden.

Als Antwort auf die dritte Aussage „Ich würde gerne an weiteren Modulen des InfoSphere teilnehmen.“ ergibt sich eine nahezu Gleichverteilung zwischen den fünf Antwortalternativen (siehe Abb. 3 c)). Auch hier konnten wieder geschlechtsspezifische Unterschiede festgestellt werden ($p < .01$). Obwohl ihnen der Modulbesuch weniger gut gefallen hat, haben Jungen ($M=2,88$, $s=1,397$) ein signifikant größeres Interesse am Besuch weiterer Module im Schülerlabor als Mädchen ($M=2,99$, $s=1,255$). Auch hier zeigt sich wieder wie gefestigt der vorher ausgeprägte Wunsch einer möglichen Beschäftigung mit der Informatik ist und wie eingeschränkt ein eintägiges Modul dahingehend wirken kann.

Argument 2 - Vermittlung eines interessanten Bildes der Disziplin

Als erstes wurde hier die persönliche Einschätzung zwischen den beiden Extrema „Informatik ist interessant (1) versus uninteressant (100).“ erhoben. Dabei zeigen sich signifikante Differenzen zwischen den Angaben vor und nach dem Modulbesuch ($p < .05$). Dabei liegt der Mittelwert vorher bei $M=24,98$ ($s=26,427$) und nachher bei $M=27,29$ ($s=27,661$). Aus dieser Verschiebung lässt sich ablesen, dass die Informatik im Schnitt nach dem Modul leicht „uninteressanter“ wahrgenommen wird. Spannend sind die signifikanten Geschlechterunterschiede ($p < .05$). Bei Jungen hat sich die Bewertung

um 3,90 Punkte Richtung „uninteressant“ verschoben, wohingegen sie sich bei Mädchen um 0,10 Punkte Richtung „interessant“ verändert hat, was unter anderem darin zu begründen ist, dass das Interesse von den Jungen im Vorfeld bereits extrem hoch angegeben wurde (Jungen vorher $M=19,11$, Mädchen vorher $M=33,89$). Obwohl dies nicht beabsichtigt war, kann der Rückgang des Interesses bedeuten, dass das vorherige Bild der Disziplin positiv verzerrt war und nun ein realistisches Bild entstanden ist.

Bei der zweiten Aussage ging es darum, ob die Informatik eher langweilig (1) oder spannend (100) ist. Für die gesamte Stichprobe konnten keine signifikanten Veränderungen durch die Module bemessen werden. Allerdings zeigte sich, dass die Antworten signifikant vom Geschlecht der Befragten abhängen ($p<.05$). Bei Jungen veränderte sich der Mittelwert um 2,33 Punkte Richtung „langweilig“, bei Mädchen entgegengesetzt um 2,29 Punkte Richtung „spannend“. Dabei gaben die Jungen im Pretest wesentlich positivere Werte ($M=77,51$) an als die Mädchen ($M=60,25$).

Auch bezüglich der Aussage „Informatik ist eintönig (1) versus abwechslungsreich (100).“ ergaben sich insgesamt keine signifikanten Differenzen zwischen den beiden Erhebungszeiträumen. Allerdings zeigte sich hier eine signifikante Abhängigkeit vom besuchten Modul. Durch den Besuch des Moduls „Scratch“ veränderte sich der Mittelwert um 2,19 Punkte in Richtung „abwechslungsreich“, nach dem Modul „Erste App“ um 0,39 Punkte ebenfalls in Richtung „abwechslungsreich“, nach dem Modul „InfoSphere goes Android“ jedoch um 0,29 Punkte in Richtung „eintönig“.

Argument 3 - Passung zu den Erwartungen der Programmierneinsteiger

Bezüglich dieses dritten Argumentes sollte die Zustimmung zu der expliziten Aussage „Das Modul entsprach meinen Erwartungen.“ angegeben werden. Hierbei ergab sich folgende Verteilung in der gesamten Stichprobe: 115x „trifft voll zu“ (1), 102x „trifft zu“ (2), 130x „weder noch“ (3), 70x „trifft eher nicht zu“ (4), 54x „trifft gar nicht zu“ (5). Auch hier ergaben sich wieder signifikante Geschlechterunterschiede ($p<.05$). Dabei wurde die Erwartung der Mädchen mit $M=2,59$ ($s=1,180$) stärker erfüllt als diese der Jungen $M=2,73$ ($s=1,377$).

Weiter wurde analysiert, welche Erwartungen diejenigen mit ins Schülerlabor brachten, welche die deutlichsten Verschiebungen hin zu „Informatik ist uninteressant.“, „Informatik ist langweilig.“ und „Informatik ist eintönig“ verzeichneten. Die Erwartungen dabei entsprechen inhaltlich dem besuchten Modul, also beinhalteten z. B. „App-Programmierung“, „etwas Neues lernen“ und „Spaß haben“. Eine Vermutung für die negative Interessenentwicklung könnte sein, dass diejenigen enttäuscht wurden, die erwarteten, dass ihnen der Freiraum einer komplett eigenen Idee für die App bzw. das Spiel gelassen würde, was mehrfach in den Durchführungen als Kritik geäußert wurde.

Die Auswertung der Verbesserungsvorschläge ist leider wenig zielführend, da meist „keine“ oder unklare Angaben wie „interessanter gestalten“ genannt wurden.

Argument 4 - Einfluss auf die wahrgenommene Schwierigkeit der Disziplin

Mittels der im Pre- sowie im Posttest zu wertenden Aussage „Informatik ist schwierig (1) versus einfach (100).“ wird das vierte Argument beleuchtet. Hier zeigt sich eine signifikante Veränderung zwischen den beiden Erhebungszeitpunkten ($p < .05$). Die Informatik wird nach dem Modul mit $M=48,2145$ ($s=24,91$) schwieriger eingeschätzt als im Vorfeld $M=51,3316$ ($s=25,00$). Auch wenn es keine signifikanten Unterschiede zwischen den Teilgruppen gibt, zeigt ein Blick in die Daten, dass diese Veränderung bei Mädchen deutlicher stattfindet. Dieses Ergebnis bestärkt den Verdacht, dass einige Jugendliche mit einem positiv verzerrten Bild der Informatik das Schülerlabor besuchen und von den Herausforderungen der Programmierung überrascht wurden.

Argument 5 - Vermittlung eines kreativen Berufsbildes „Informatiker-in“

Mit Blick auf das fünfte Argument wird die Zustimmung zu zwei Aussagen auf einer Skala von „stimme gar nicht zu“ (1) bis „stimme voll zu“ (11) erhoben. Dabei zeigt sich ein signifikanter Unterschied bei der Aussage „Informatiker-innen müssen kreativ sein.“ ($p < .05$). Nach dem Modul wird dieser Aussage mit $M=7,9427$ ($s=2,66$) signifikant stärker zugestimmt als vorher $M=7,4268$ ($s=3,05$). Mittels eines grafischen Programmereinstiegs gelingt es den Schülerinnen und Schülern zu verdeutlichen, dass Softwareentwicklung ein kreativer Prozess sein kann. Dabei sind die Veränderungen von keiner trennenden Variablen signifikant abhängig. Auch bei der Auswertung der Pre-Test-Daten zeigten sich keine spezifischen Unterschiede unter den Teilgruppen.

Zum Abschluss wurde die Aussage „Informatiker-innen kreieren viel Neues.“ evaluiert. Hierbei zeigten sich im Vorfeld wie auch im Vergleich zwischen den beiden Erhebungszeiträumen keine signifikanten Unterschiede.

5 Reflexion & Ausblick

Insgesamt hat die quantitative Auswertung der 471 Pre-Posttest-Datensätze gezeigt, dass die Module im Schülerlabor zum Einstieg in die grafische Programmierung mittels Scratch und dem App Inventor nicht unbedingt zu einem positiveren Bild der Informatik beitragen, aber vermutlich ein realistischeres Bild erzeugen.

Argument 1 - Begeisterungsfähigkeit: Die Auswertung zeigt, dass die Module den Mädchen signifikant besser gefallen haben, dennoch sind es die Jungen, die angeben sich in Zukunft eher mit Informatik beschäftigen zu wollen. Weiter zeigt sich, dass das anspruchsvollste Modul am schlechtesten bewertet wird, was ein Indiz dafür ist, dass Überforderung zur Abnahme der Begeisterung führt.

Argument 2 - Vermittlung eines interessanten Bildes der Disziplin: Bei einem Teil der Jungen, veränderte sich das Bild der Informatik teils negativ. Dieser Rückgang des Interesses kann auch in einem im Vorfeld extrem positiven, unter Umständen gar verzerrten Bildes der Informatik begründet sein. Das Bild der Informatik veränderte sich bei Jungen in Richtung „langweilig“, bei Mädchen jedoch exakt entgegengesetzt in

Richtung „spannend“. Somit kann davon ausgegangen werden, dass sich die eigene Vorstellung mancher Besucherinnen und Besucher hin zu einer realistischeren Einschätzung entwickelt hat. Weiter haben sich die zuvor stark ausgeprägten Unterschiede zwischen den Geschlechtern leicht angeglichen.

Argument 3 - Passung zu den Erwartungen der Programmierneinsteiger: Die Auswertung des Pretests zeigte, dass die Erwartungen zu den Inhalten der Module passen, es jedoch Enttäuschungen bezüglich der gewünschten Freiheitsgrade innerhalb der Projekte gab. Insgesamt gaben Mädchen deutlicher als Jungen an, dass die Module ihre Erwartungen erfüllt hätten. Dies könnte eine Begründung für die negativen Verschiebungen bzgl. der wahrgenommenen Interessantheit der Disziplin bei den Jungen sein.

Argument 4 - Einfluss auf die wahrgenommene Schwierigkeit der Disziplin: Die Tatsache, dass die Informatik nach einem Programmierneinstieg als schwieriger bewertet wird, legt nahe, dass einige Lernende mit einem verzerrten Bild der Informatik das Schülerlabor besuchten und von den Anforderungen überrascht wurden.

Argument 5 - Vermittlung eines kreativen Berufsbildes „Informatiker-in“: Der grafische Programmierneinstieg zeigt die Softwareentwicklung als kreativen Prozess und führt zur signifikanten Überzeugung, dass Informatiker-innen kreativ sein müssen.

Da der Einstieg in die Programmierung dank passender Werkzeuge zukünftig bereits im Grundschulalter stattfinden kann, soll die Evaluation auf diese jüngere Zielgruppe ausgeweitet werden. Dazu werden auch Apps zum grafischen Programmierneinstieg wie „Scratch Junior“ und Hardware wie LEGO Wedo, Ozobot und Calliope hinsichtlich ihres Potentials bezüglich der Schülervorstellungen über Informatik beleuchtet.

Literaturverzeichnis

- [Be15] Bergner, N.: Konzeption eines Informatik-Schülerlabors und Erforschung dessen Effekte auf das Bild der Informatik bei Kindern und Jugendlichen. Dissertation, Aachen, 2015.
- [Ma10] Maloney, J. et al.: The Scratch Programming Language and Environment. In ACM Transactions on Computing Education, 2010, 10; 16: 1–15.
- [Ut10] Utting, I. et al.: Alice, Greenfoot, and Scratch -- A Discussion. In ACM Transactions on Computing Education, 2010, 10; S. 1–11.
- [Wo11] Wolber, D.: App inventor and real-world motivation. In (Cortina, T. J. Hrsg.): Proceedings of the 42nd ACM technical symposium on Computer science education. ACM, New York, NY, 2011; S. 601.

Modularisierung im Informatikunterricht aus lernpsychologischer Perspektive

Johannes Fischer, Arno Pasternak¹

Abstract: Objektorientiertes Modellieren und Programmieren ist eine weitverbreitete Technik, um informatische Prinzipien wie Abstraktion, Automation und Modularisierung umzusetzen. Letztgenannte Begriffe gehören zweifelsohne zu den wichtigsten Konzepten des *Informatischen Denkens* nach Jeannette Wing [Wi06], und ihre Vermittlung ist aus einem allgemeinbildenden Informatikunterricht nicht wegzudenken. Unklar ist nur der zu beschreitende Weg, um diese Kompetenzen zu erlangen. Kölling und Rosenberg [KR01] geben hierfür einige Ratschläge, die sich insbesondere durch die Verwendung *großer* Projekte von Anfang an auszeichnen. In diesem Beitrag wird dafür plädiert, statt einer konkreten Technik wie der Objektorientierung die Konzepte Abstraktion, Automation und insbesondere Modularisierung in den Fokus zu nehmen und diese altersgerecht mit einer geeigneten Sprache zu unterrichten.

Keywords: Informatikunterricht, OOM, OOP, Module, Lernpsychologie, Cognitive Load Theory

1 Informatische Bildung an Schulen

Heymann hat 1996 überzeugend dargelegt, was unter *Allgemeinbildung* heute verstanden werden soll [He96]. Entsprechend ist der Allgemeinbildungscharakter der Informatik unter Fachdidaktikern unbestritten [Wi03, SS11, Pa13]. Diese Bedeutung wird zusätzlich durch die Darstellung von *Jeannette Wing* deutlich, die 2006 in einem kurzen Beitrag die Besonderheit des *informatischen Denkens* dargestellt hat [Wi06].

Es versteht sich dabei von selbst, dass in der Informatik im Schulunterricht als Allgemeinbildung in erster Linie grundlegende Strukturen und Konzepte anstatt die Nutzung bestimmter aktueller Werkzeuge in Form konkreter Informatiksysteme wie beispielsweise eine Textverarbeitung oder eine konkrete Datenbankimplementierung vermittelt werden sollen. Ebenso gilt, dass auch bei der Verwendung informatikinterner Werkzeuge der Schwerpunkt auf die Vermittlung von Konzepten und Prinzipien und nicht auf beispielsweise die Anwendung einer oder mehrerer Programmiersprachen und mit denen in diesen Sprachen realisierbaren Ideen gelegt werden muss.

1.1 Konzepte der Informatik

Welches sind die grundlegenden Konzepte der Informatik? Ausgehend von ihren ersten Feststellungen schreibt Wing 2008 [Wi08]: „*Abstraction and automation*“ und etwas später in anderer Formulierung:

¹ TU Dortmund, Fakultät für Informatik, Otto-Hahn-Str. 14, 44227 Dortmund,
{Johannes.Fischer,Arno.Pasternak}@cs.tu-dortmund.de

„Computing³ is the automation of our abstractions“. Diese kurze und knappe Beschreibung verlangt natürlich nach einer genaueren Begriffsanalyse. Erst die Kombination aus *abstrakter* Betrachtung und Modellbildung der realen Welt und deren Umsetzung in eine durch Maschinen *ausführbare* Implementierung macht den besonderen Charakter der Informatik aus und hat durch die dafür notwendigen geistigen Vorstellungen und Bemühungen den intellektuellen Horizont der Menschen wesentlich erweitert und durch die dadurch erzeugten Produkte die Welt real verändert.

Der Kerngedanke der Aussagen von Wing lässt sich als Konzept folgendermaßen beschreiben: Durch *Abstraktion* wird als Abbild der realen Welt ein Modell erschaffen. Dies stellt aber erst eine *statische* Beschreibung der (Modell-)Welt dar. Erst durch die Beschreibung von *abstrakten Handlungen* wird ermöglicht, dass bei Vorhandensein einer entsprechenden Implementierung ein Ergebnis erzielt werden kann, das (hoffentlich) bei der Lösung eines Problems hilft. Durch das Ausführen dieser Implementierung wird aus dem statischen ein *dynamisches* Modell.

Dieses dynamische Modell wird auf einer (konkreten) Maschine ausgeführt. Diese Maschine „versteht“ leider keine natürlichen Sprachen. Die künstlichen Sprachen der Maschinen sind eine Anpassung an technische Gegebenheiten und stellen eine eigene intellektuelle Struktur dar. Diese Programmiersprachen müssen erlernt und durch Übungen gefestigt werden. Es ist eine intellektuell hoch zu wertende Leistung, ein Modell in einer konkreten Programmiersprache zu „kodieren“ und dieses Programm zu testen und zu korrigieren. Bei der Umsetzung eines Modelles in eine derartige Sprache sind gerade sogenannte *Novizen* erheblich intellektuell gefordert. Entsprechend ist der Aufwand nicht zu unterschätzen, der im Unterricht beim Erlernen einer Programmiersprache nötig ist. Dieser hohe Aufwand darf aber nicht dazu führen, auf die Automatisierung durch Programmierung im Unterricht zu verzichten.

Die beiden Aspekte des *informatischen Denkens* — *Abstraktion und Automation* — müssen in einem Schulfach Informatik (mit didaktischer Reduktion) gleichermaßen abgebildet werden, wenn es seinem Charakter gerecht werden will. Betrachten wir also die beiden Aspekte etwas genauer.

2 Abstraktion

Um ein Problem der realen Welt mit einer Maschine — dem Computer — zu bearbeiten, muss das Problem in seine Bestandteile zerlegt und strukturiert analysiert werden. Daraus folgt im Endeffekt eine Modularisierung und eine strukturierte Programmierung.

Diese *Modellierung* zu einer reduzierten abstrakten Welt erhalten wir durch Entwicklung von *Modulen*. Im *Informatik-Duden* werden derartige Module beschrieben als:

- „Er⁴ ist logisch oder funktional in sich abgeschlossen.

³ Mit *Computing* ist in der englischen Fachsprache *Informatik* gemeint.

⁴ gemeint ist hier: das Modul als *der* Baustein

- Wie er arbeitet oder implementiert ist, braucht außen nicht bekannt zu sein [...].
- Er besitzt klar definierte Schnittstellen nach außen.

[...] Ein System ist *modular* aufgebaut, wenn es aus abgrenzbaren Einheiten zusammengesetzt ist und wenn diese Einheiten einzeln ausgetauscht, verändert oder hinzugefügt werden können, ohne dass andere Teile des Systems hierdurch beeinflusst werden oder das System arbeitsunfähig wird.“ [CS03, S.414]

Abstraktion der Daten Diese Definition macht deutlich, wie zentral die Idee der Modularisierung in der Informatik ist. Beschreibt ein derartiges Modul Daten aus der Realität, so sprechen wir von einer *abstrakten Datenstruktur* oder einem *abstrakten Datentyp* [Po84, S.85ff,S.154ff]⁵ Da die Beschreibung von Daten eine zentrale Aufgabe im Rahmen der Modellierung ist, ist das Erlernen der Entwicklung von derartigen Modulen ein wichtiger Teil im Rahmen von Lehrplänen für das Schulfach Informatik.

Abstraktion der Abläufe Eine systematische Abstraktion der Abläufe führt zur *strukturierten Programmierung*, die im Informatikstudium beschrieben wird als: „Programmiermethode, bei der das vorgegebene Problem in Teilprobleme und die Beziehungen zwischen diesen Teilproblemen(Schnittstellen) zerlegt wird. [...]“ [CS03, S.643] Je nach Programmierparadigma werden bei der imperativen Programmierung *Kontrollstrukturen* oder bei der funktionalen Programmierung *rekursive Funktionen und Prozeduren* verwendet.

Abstraktion im Unterricht Beide Formen der Abstraktion müssen im Informatikunterricht gleichermaßen für eine Problemlösung vermittelt werden. Eine ausschließliche Abstraktion der Daten führt zu einer Beschreibung der statischen Welt, vergleichbar mit dem Kunstunterricht, eine ausschließliche Abstraktion der Abläufe führt de facto zu einer Variante des Mathematikunterrichtes. *Schubert* und *Schwill* machen daher bei ihrer beispielhaften Vorstellung des objektorientierten Vorgehens deutlich: „Es soll deutlich werden, dass die traditionellen, grundlegenden Konzepte der strukturierten Programmierung unverzichtbarer Lerngegenstand sind, auch wenn objektorientierte Lösungen von den Schülern konstruiert werden. Folgende Teilziele werden angestrebt: Die Schüler können das Konzept der Modularisierung anwenden. ...“ [SS11, S.157] Sie berufen sich dabei auf *Böszörményi*, der richtigerweise feststellt, dass „das Konzept der Modularisierung viel grundlegender [ist] als das Konzept der Objektorientierung“ [Bö01, S.15]. Folgt man *Wirth*, handelt es sich dabei noch nicht einmal um ein neues, eigenständiges Konzept, sondern nur um eine neue Technik: „Noch ein Wort zur Objektorientierung: [...] Man lernt neue Programmiertechniken – aber auf dem Boden des bisher Mitgebrachten.“ [Wi91, S.60,61]

Objektorientierte Technologie im Unterricht In der Ausbildung werden heute häufig objektorientierte Sprachen verwendet. Dass modulare Modellierung und Programmierung Objektorientierung nicht voraussetzt, wird damit den Lernenden nicht immer deutlich. Wir haben beobachtet, dass selbst Informatik-Studierende nur die objektorientierte Darstellung kennen und daher glauben, dass modulares Programmieren mit Objektorientierung gleichzusetzen ist. So verwundert es nicht, dass dieser Eindruck auch in einem Schulbuch vertreten wird:

⁵ Eine *abstrakte Datenstruktur* definiert nur ein Exemplar, ein *abstrakter Datentyp* definiert eine Menge von gleichen abstrakten Datenstrukturen. [Po84, S.100]

„Vor allem in Zeiten des Internets und der Verbreitung von Software in allen Lebensbereichen bietet die OOP große Vorteile:

- Wiederverwendbarkeit von schon programmierten Elementen
- Aufteilung überschaubarer Einzelteile
- Erweiterung durch Schnittstellen“ [KT10, S.15]

Diese unvollständige fachliche Beschränkung der Modularisierung auf Objektorientierung verführt leicht zu fragwürdigen didaktischen Entscheidungen ⁶.

3 Automation

Aus der abstrakten Beschreibung von Daten und Abläufen soll ein Programm in einer konkreten Programmiersprache erstellt werden, das auf einer Maschine ablauffähig ist. Die Erstellung eines solchen Programmes ist aber mitnichten einfach oder trivial. Die gewählte Programmiersprache stellt entsprechend der Syntax bestimmte grundsätzliche Datentypen und Ablaufstrukturen als Bausteine zur Verfügung. Um vom abstrakten Modell zum konkreten Programm zu gelangen, muss ein *Graben* zwischen den abstrakten Ideen und den konkreten Strukturen der jeweiligen Programmiersprache überwunden werden.

Der Modellierungsgraben Dieser Graben wird am Beispiel des Kernlehrplanes für die gymnasiale Oberstufe in NRW [MS13, S.21] deutlich, wenn die (verkürzte) Aufzählung um die Darstellung dieses Grabens erweitert wird:

„Die Schülerinnen und Schüler
<ul style="list-style-type: none">• ermitteln bei der Analyse einfacher Problemstellungen Objekte, ihre Eigenschaften, ihre Operationen und ihre Beziehungen (M),• modellieren Klassen mit ihren Attributen, ihren Methoden und Assoziationsbeziehungen (M),
GRABEN
<ul style="list-style-type: none">• ordnen Attributen, Parametern und Rückgaben von Methoden einfache Datentypen, Objekttypen oder lineare Datensammlungen zu (M)“

Beispielhaft ist es für Schüler überhaupt nicht selbstverständlich, das Sortieren einer Menge von Karten mit einem Feld oder einer Liste zu modellieren. Die Darstellung in den meisten Schulbüchern suggeriert allerdings diese scheinbare Selbstverständlichkeit, die von vielen Lehrern auch ohne große didaktische Reflexion den Schülern übergestülpt wird. Beeinflusst der Lehrer die Schülerinnen und Schüler nicht, so modellieren diese ohne Kenntnis der Datenstruktur Feld oder Liste wahrscheinlich einzelne Karten.

Ebenso wie in der obigen Darstellung für die Daten angegeben, ist auch ein entsprechender Graben für die algorithmische Abstraktion zur Programmiersprache vorhanden. Jeder Lehrer weiß, wie lange es dauert, bis die Schülerinnen und Schüler beispielsweise die Iteration über ein Feld als geeignetes programmiersprachliches Konstrukt einer Problemstellung erkennen und auch anwenden können.

⁶ In der Ausgabe von 2014 ist die obige Darstellung so nicht mehr enthalten [KL14, S.20]

Ein Graben kann überwunden werden, wenn man von der einen Seite auf die andere Seite springen kann. Dieses muss aber erlernt werden. Dabei ist ein Sprung über einen Graben von der einen Seite zumeist genauso kompliziert wie von der anderen Seite. Das bedeutet hier: Wenn es mir als Schüler gelingt, die Wirklichkeit zu modellieren, muss ich zusätzlich lernen, wie diese Modelle in der gewählten Programmiersprache abgebildet werden können. Kann ich mich zuerst in dieser Programmiersprache für kleine Problemstellungen adäquat bewegen, muss ich zusätzlich lernen, die Wirklichkeit in diese mir bekannten Strukturen „herunterzubrechen.“

Unterrichtliches Vorgehen Notwendig sind also Kenntnisse der Datenrepräsentation und Ablaufsteuerung. Beide Aspekte müssen erlernt und geübt werden. Welcher davon im unterrichtlichen Verlauf sinnvollerweise wann im Vordergrund steht, ist nicht offensichtlich. Diese Einschätzung wird durch die Untersuchung von *Ehler* gestützt, der in seiner Studie verglichen hat, ob eine objektorientierte Sichtweise gegenüber der prozedural-imperativen Vorgehensweise im Anfangsunterricht Vorteile hat: „Es kann das Vorgehen gewählt werden, welches der Lehrer bzw. Dozent für das richtige hält.“ [Eh12, S.218]

4 Erfahrungen aus der Schulpraxis

In den letzten Jahren hat sich in der Schule die Sprache *Java* gegenüber beispielsweise (*Objekt-)**Pascal* durchgesetzt. *Java verlangt* das Arbeiten mit Klassen. Die Technik der Objektorientierung und damit auch die Sprache *Java* ist zweifelsohne für die Implementierung großer Programmsysteme sehr gut geeignet. Daraus kann abgeleitet werden, auch in der Schule mit diesen Ideen und Techniken der Praxis zu beginnen. Entsprechend formulieren Schubert und Schwill richtigerweise: „Die Einführung in die Informatik kann mit objektorientiertem Modellieren (OOM) und Programmieren (OOP) [. . .] beginnen. Zwingend ist das nicht.“ [SS11, S.157] Eine derartige Entscheidung hat allerdings auch Nachteile. Klassen repräsentieren *Datenkapseln*. „Programmsysteme, die mit Datenkapseln arbeiten, enthalten deshalb fast immer mehr Prozeduren als konventionelle Programmsysteme.“ [Po84, S.99] Gerade bei kleineren Programmen, die im Anfangsunterricht vermehrt auftreten, stellen diese oft einen *Overhead* dar.

OO-Guidelines Diese Grundproblematik objektorientierter Systeme und zusätzlich speziell für Anfänger schwer durchschaubare syntaktische Strukturen der Sprache *Java* verleiten Kölling und Rosenberg zu Vorschlägen, wie ein Anfangsunterricht mit der Sprache *Java* erfolgversprechend sein könnte [KR01]. Der relative *Overhead* sei kleiner, je größer das System ist. Also folgern sie, daß der Unterricht mit einem großen System beginnen sollte.

Einige dieser von Kölling und Rosenberg empfohlenen *guidelines* beziehen sich auf die Ideen der Objektorientierung (z.B. „*use 'large' projects*“, „*objects first*“), andere auf die Problematik der konkret verwendeten Sprache *Java* (z.B. „*don't start with 'main'*“, „*be careful with the user interface*“). Es werden damit zwei völlig unterschiedliche didaktische Fragestellungen in diesen Vorschlägen vermischt.

Umsetzung im Unterricht Wir haben uns gefragt, ob die sprachspezifischen Aspekte von Java tatsächlich das Erlernen der Objektorientierung erschweren und haben in einem kleinen Unterrichtsprojekt von ca. 10 Stunden eine erste dahingehende Analyse versucht. In einem 12. Jahrgang einer Gesamtschule wurden zwei etwa gleich große Kurse eingerichtet (je ca. 15 Schüler). In einem Kurs wurde mit Java unterrichtet, in dem anderen Kurs mit der Scriptsprache *Groovy*⁷, die die sprachspezifischen Probleme von Java teilweise umgehen lässt. Wie weiter unten beschrieben wird, erwiesen sich hier die Sprachunterschiede aber nicht als problematisch.

In beiden Kursen wurde ein Schülerverwaltungsprogramm in der jeweiligen Sprache mit ca. 500 Zeilen verwendet. Entsprechend den Vorstellungen von Kölling und Rosenberg waren in diesem Programm mehrere Klassen enthalten, zumindest teilweise wurden mehrere Instanzen dieser Klassen verwendet. Die Schüler haben diesen Programmtext gelesen, getestet, kleine Veränderungen vorgenommen, neue Instanzen erzeugt und auch einzelne Operationen zielgerichtet verändert.

Erwartungen Erwartet hatten wir, dass sich die Schülerinnen und Schüler in der Groovy-Gruppe aufgrund der einfacheren Syntax leichter tun würden. Dies spielte allerdings in dieser Phase kaum eine Rolle, da das Programm als Ganzes vorlag und nur an einzelnen Stellen, die sich in Java und Groovy textuell kaum unterschieden, verändert wurde. Allerdings zeigte sich bei der Analyse eine andere, viel bedeutsamere Problematik.

Das Ziel des Ansatzes von Kölling und Rosenberg ist es, dass die Lernenden durch das explorative Erarbeiten eines nicht zu kleinen Programmes die Vorteile des objektorientierten Vorgehens selbst erkennen und die Begrifflichkeiten anhand des Beispiels deutlich werden. Um dies zu überprüfen, fragten wir zu Beginn des Schuljahres und nach dieser Unterrichtsphase schriftlich die Schüler, was sie sich unter Begriffen wie beispielsweise *Klasse* und *objektorientierter Programmierung* vorstellen. Es muss dabei angemerkt werden, dass die Begriffe im Lauf des Unterrichts bei der Analyse selbstverständlich erläutert, erarbeitet und auch verschriftlicht wurden. Daher gingen wir davon aus, dass bei diesen rein reproduktiven Fragen zumindest die erlernten Definitionen angegeben werden würden.

Erste Ergebnisse Das Ergebnis hat uns völlig überrascht. Nicht eine einzige Antwort in beiden Gruppen war zielführend. Eine Antwort wie „*Eine Klasse wird mit Attributen gefüllt*“ kann dabei noch als Ansatz einer korrekten Idee angesehen werden, aber Antworten wie „*Das ist das Programmieren mit Hilfe von Datenbanken*“ oder „*Unter objektorientierter Programmierung kann man mit speziellen Objekten programmieren*“ waren häufiger. Entsprechend der von uns gestellten Ansprüche an den Lernzuwachs der Schülerinnen und Schüler war das Ergebnis vernichtend. Die Ideen des Konzepts der Modularisierung – hier mit der Technik der Objektorientierung realisiert – waren offensichtlich nicht erlernt und verstanden worden, sodass selbst eine einfache reproduktive Wiedergabe der Begriffe von den Schülern nicht geleistet werden konnte.

Da keine Vergleichsdaten aller Schüler wie in dieser Befragung von früheren Kursen mit Java bzw. Groovy mit einem anderen didaktischen Vorgehen vorliegen, ist es nicht ausge-

⁷ <http://www.groovy-lang.org>, letzter Zugriff: 6.1.2017

schlossen, dass auch in der Vergangenheit keine besseren Ergebnisse erzielt worden sind. Dagegen spricht allerdings folgende Beobachtung: In der folgenden Klausur, an der allerdings nur sechs der insgesamt 28 Schülerinnen und Schüler teilgenommen haben, wurden wie in vergleichbaren Klausuren der vergangenen Jahre diese Begriffe erneut erfragt. Diese Aufgabe ist normalerweise als reproduktive Aufgabe ein „Punktlieferant“ für die Klausurteilnehmer. Bis auf eine Lösung waren die Antworten wieder nicht ausreichend.

Die Kurse in den früheren Jahren sind didaktisch mit deutlich kleineren Beispielen an die Thematik herangeführt worden. Die Antworten der Schüler, die vor einem Jahr an der Klausur teilgenommen haben, waren deutlich besser.

Da wir immerhin noch auf 8 der damals geschriebenen 22 Klausuren Zugriff hatten, konnten wir die Antworten an diesen Klausuren verifizieren. Zusätzlich liegen ähnliche Daten aus einem Kurs des 11.(!) Jahrganges aus dem Jahre 2010 vor. Dieser Kurs hatte allerdings ein etwas anderes Curriculum (und wurde in (Objekt-)Pascal unterrichtet). Die Schülerantworten stellten den Lehrer im Gegensatz zu den Antworten der beiden Kurse im aktuellen Jahr im Wesentlichen zufrieden.

Spekulativer Vergleich dieser Ergebnisse Es handelt sich bei diesen Zahlen nicht um eine systematische Untersuchung. Die Ergebnisse und der Vergleich mit früheren Kursen zeigen trotzdem, dass die Schülerinnen und Schüler beim Vorgehen nach Kölling und Rosenberg die Modularisierung und deren konkrete Umsetzung mit der Objektorientierung nicht verstanden haben. Selbst die unter Groovy niedrigere Syntax-Hürde konnte dies nicht verhindern. Die Schülerinnen und Schüler aus den Kursen vor einem Jahr und dem Jahre 2010 konnten die Begriffe der OOP im Wesentlichen wiedergeben.

Aufgrund der schwierigen Vergleichbarkeit der Materialien und der unterschiedlichen Kursverläufe dürfen diese Ergebnisse nicht überbewertet werden. Deutlich wird aber, dass dringender Bedarf nach empirischen Untersuchungen besteht.

5 Lernpsychologische Sicht

Die aus fachlicher Sicht überzeugende Position von Kölling und Rosenberg hatte uns angeregt, in diesem Schuljahr nach dieser Konzeption vorzugehen. Das Ergebnis war ernüchternd. Es erscheint uns naheliegend, dass hier lernpsychologische Gründe für das Scheitern verantwortlich sind.

In den letzten ca. zwanzig Jahren hat es erhebliche Fortschritte in der Lernpsychologie gegeben. Mit diesen ist es möglich, die Funktionsweise des Lernens im menschlichen Gehirn zu verstehen. Eine wesentliche Theorie stellt die *Cognitive Load Theorie (CLT)* [PRS03] dar, die in inzwischen vielfältigen Untersuchungen bestätigt wurde. Nach dieser Theorie lässt sich die Arbeitsweise des Gehirns als das Zusammenspiel zwischen einem *Arbeitsgedächtnis* und einem *Langzeitgedächtnis* vorstellen. Das nur gering belastbare Arbeitsgedächtnis nutzt dabei das Langzeitgedächtnis als Wissens- und Kompetenzspeicher. Es lassen sich aber nur wenige neue Eindrücke mit diesem Speicher gleichzeitig verarbeiten. Daher ist es sinnvoll, in einen Lernprozess zu einem bestimmten Zeitpunkt nur we-

nige neue Elemente einzugliedern. Die empirischen Untersuchungen von *Hattie* [HBZ13] und die neurowissenschaftlichen Begründungen von ihm und *Yates* bestätigen dies auf eindrucksvolle Weise [Ha15, S.107ff]. Für die Informatik bedeutet dies, dass gerade im Anfangsunterricht das *bottom-up*-Prinzip oft dem *top-down*-Prinzip vorzuziehen ist.

Mit diesen neurowissenschaftlichen Überlegungen können die Theorien von *Piaget* und *van der Hiele* aus der Mitte des letzten Jahrhunderts begründet werden [PI93, HHG78]. Entsprechend dieser Erkenntnisse hat die Gruppe der *Neo-Piagisten* wie beispielsweise *Lister* [Li16] diese Theorien weiterentwickelt. Für den Informatikunterricht speziell folgert *Lister* [Li16, S.9ff] ein intensives Arbeiten an und mit dem Quellcode eines Programmes. Entsprechendes wurde auch einige Jahre zuvor von *Shaffer*, *Doub* und *Tuovinen* aus der CLT-Theorie gefolgert [Sh03]. Eigene Untersuchungen aus dem Anfangsunterricht mit einer Programmiersprache bestätigen dies [Pa16].

Nach der CLT-Theorie ist das aus Sicht der Objektorientierung sinnvolle Vorgehen mit dem Erarbeiten eines größeren Softwareprojektes als Einführung in die Modellierung und Programmierung eine Überlast des lernenden Gehirns und muss daher im Allgemeinen scheitern. Es ist daher nach einem anderen Vorgehen zu suchen.

6 Eine mögliche Alternative

Datenabstraktion bleibt das Ziel. Die objektorientierte Programmierung ist eine Möglichkeit, die offensichtlich im Unterricht der allgemeinbildenden Schulen nicht auf jedem Weg — wie gezeigt — zum Ziel führt. Auch wenn die OOM und die OOP als programmier-technisches Vorgehen angestraft wird, ist es sinnvoll, den Weg dorthin anders zu gestalten. Das lässt sich nur mit Programmiersprachen realisieren, die verschiedene Umsetzungen der Datenabstraktion ermöglichen. Gut ist es, wenn die verschiedenen Techniken in der Darstellung sich nicht groß unterscheiden, sodass ein Wechsel der Darstellung aus didaktischer Sicht keine unnötigen syntaktischen Schwierigkeiten bereitet. Hierzu sind allerdings noch empirische Untersuchungen nötig.

Module und Klassen in einer geeigneten Sprache Im Folgenden wird ein Beispiel bezüglich der Datenabstraktion in der Programmiersprache Tc1 [F112] vorgestellt. Es geht hier nicht um diese konkrete Sprache, sondern um das Prinzip. In Tc1 sind alle Formen der Datenabstraktion möglich. Die Nutzung der entsprechenden Techniken weist in der Darstellung kaum Unterschiede auf.

Als Beispiel diene das Einfügen in *binäre Bäume*. Ohne Nutzung von Modulen bzw. mit importierten Modulen lautet der Aufruf: `fuege_ein mein_baum $info`. Bei der Nutzung von Klassen ändert sich beim Aufruf `mein_baum fuege_ein $info` nur die Reihenfolge in der Anweisung. Bei Verwendung einer abstrakten Datenstruktur statt eines abstrakten Datentypes vereinfacht sich die Anweisung sogar noch: `fuege_ein $info`.

Die fast gleichartige Darstellung bei der Verwendung von Modulen oder Klassen für abstrakte Typen macht deutlich, dass das Denken in diesen Strukturen sich programmiersprachlich ohne große Brüche in der jeweils gewünschten Form umsetzen lässt. Das

ermöglicht die Umsetzung einer abstrakten Struktur in die technische Form einer Programmiersprache, die zu dem unterrichtlichen Zeitpunkt entsprechend dem Stand der Schüler angemessen ist und kann eine Überforderung an technischem und begrifflichem Overhead im Anfangsunterricht verhindern helfen. Auf diese Weise können die Schülerinnen und Schüler sanft ohne intellektuelle Überforderung von der Strukturierung der Abläufe mit Prozeduren über die Gestaltung von Modulen bis zur objektorientierter Programmierung geführt werden und damit die Gedanken und Ideen der Modularisierung auf vielfältige Weise erfahren und erlernen. Das erfordert allerdings die Auswahl einer Programmiersprache für den Unterricht, die verschiedene Darstellungsmöglichkeiten gleichberechtigt ermöglicht.

Mit einem derartigen Vorgehen wird auch einem weiteren wichtigen Prinzip gefolgt, das nicht nur, aber auch in der Informatik Gültigkeit besitzt, das *Kiss-Prinzip* [Ri95]: *Keep it small and simple*. Eine Hoffnung aus der Anwendung dieses Prinzip in der Modellierung und vor allem in der Umsetzung in eine konkrete Programmiersprache ist, die Hürden für die Schüler so niedrig wie möglich zu setzen, sodass diese auch außerhalb des Unterrichts und im späteren Leben für sich selbst nicht ausschließen, ein konkretes Problem durch ein informatisches Vorgehen selber programmieren und lösen zu wollen.

7 Zusammenfassung

Ein wesentliches „Denkwerkzeug“ der Informatik ist die *Modularisierung*, die mit verschiedenen Techniken durchgeführt werden kann. Eine davon ist die *objektorientierte Modellierung und Programmierung*. Diese hat allerdings oft einen erheblichen Overhead zur Folge.

Bei einem Vergleich von Schülerleistungen verschiedener Jahrgänge zeigte sich, dass vor ein Vorgehen nach den „guidelines“ von Kölling und Rosenberg vielen Schülerinnen und Schülern große Schwierigkeiten macht. Es ist daher zu überlegen, ob es zweckmäßig ist, den Schulunterricht nicht von Beginn an mit objektorientierten Sprachen zu beginnen. Im Rahmen der didaktischen Reduktion könnte ein Vorgehen, dass den Aufbau der Elemente der Modularisierung stufenweise ermöglicht, geeigneter sein. Dieser Weg realisiert die Ideen, wie sie von *Kortenkamp u.a.* unter dem Begriff *genetischer Unterricht* [Ko09] beschrieben sind. Ein solches Vorgehen verlangt eine didaktisch begründete Auswahl der Programmiersprache, mit der diese Ideen sukzessive im Unterricht vermittelt werden.

Literaturverzeichnis

- [Bö01] Böszörményi, László: Java für Anfänger? LOG IN, (1/2001):14–19, 2001.
- [CS03] Claus, Volker; Schwill, Andreas: Duden Informatik. Dudenverlag, Mannheim, 2003.
- [Eh12] Ehlert, Albrecht: Empirische Studie: Unterschiede im Lernerfolg und Unterschiede im subjektiven Erleben des Unterrichts von Schülerinnen und Schülern im Informatik-Anfangsunterricht (11. Klasse Berufliches Gymnasium) in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First und OOP-Later) . Dissertation, Freie Universität Berlin, Berlin, 2012.

- [Fl12] Flynt, Clif: *Tcl/Tk: A Developer's Guide*. The Morgan Kaufmann Series in Software Engineering and Programming. Amsterdam, 2012.
- [Ha15] Hattie, John; Yates, Gregory C.R.; Beywl, Wolfgang; Zierer, Klaus: *Lernen sichtbar machen aus psychologischer Perspektive*. Schneider Verlag GmbH, Baltmannsweiler, 2015.
- [HBZ13] Hattie, John; Beywl, Wolfgang; Zierer, Klaus: *Lernen sichtbar machen*. Schneider Verlag GmbH, Baltmannsweiler, 2013.
- [He96] Heymann, Hans Werner: *Allgemeinbildung und Mathematik. Studien zur Schulpädagogik und Didaktik*; 13. Beltz, Weinheim [u.a.], 1996.
- [HHG78] Hiele, Pierre Marie van; Hiele-Geldorf, Dina van: *Die Bedeutung der Denkebenen im Unterrichtssystem nach der deduktiven Methode. Wege der Forschung*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1978.
- [KL14] Kempe, Thomas; Löhr, Annika: *Informatik 1*. Schöningh, Paderborn, 2014.
- [Ko09] Kortenkamp, Ulrich; Modrow, Eckart; Oldenburg, Reinhard; Poloczec, Jürgen; Rabel, Magnus: *Objektorientierte Modellierung - aber wann und wie?* LOG IN, (160/161):41–47, 2009.
- [KR01] Kölling, Michael; Rosenberg, John: *Guidelines for Teaching Object Orientation with Java*. In: *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education*. ITiCSE '01, ACM, New York, NY, USA, S. 33–36, 2001.
- [KT10] Kempe, Thomas; Tapaße, David: *Informatik 1 * Softwareentwicklung mit Greenfoot und BlueJ*. Schöningh, Paderborn, 2010.
- [Li16] Lister, Raymond: *Toward a Developmental Epistemology of Computer Programming*. In: *Proceedings of the 11th Workshop in Primary and Secondary Computing Education*. WiPSCE 2016, ACM, New York, NY, USA, S. 5–16, 2016.
- [MS13] MSW NRW: *Kernlehrplan für die Sekundarstufe II, Gymnasium/Gesamtschule in Nordrhein-Westfalen, Informatik*. Düsseldorf, 2013.
- [Pa13] Pasternak, Arno: *Fach- und bildungswissenschaftliche Grundlagen für den Informatikunterricht in der Sekundarstufe I*. Dissertation, Westfälische Wilhelms-Universität, Münster, 2013.
- [Pa16] Pasternak, Arno: *Contextualized Teaching in the Lower Secondary Education Long-term Evaluation of a CS Course from Grade 6 to 10*. In: *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. SIGCSE '16, ACM, New York, NY, USA, S. 657–662, 2016.
- [PI93] Piaget, Jean; Inhelder, Bärbel: *Die Psychologie des Kindes*. dtv-Taschenbücher dialog und praxis. Dt. Taschenbuch-Verlag, Stuttgart, 1993.
- [Po84] Pomberger, Gustav: *Softwaretechnik und Modula 2*. Hanser Fachbuchverlag, München, 1984.
- [PRS03] Paas, Fred; Renkl, Alexander; Sweller, John: *Cognitive Load Theory and Instructional Design: Recent Developments*. *Educational Psychologist*, 38(1):1–4, 2003.
- [Ri95] Rich, Ben R.: *Clarence Leonard (Kelly) Johnson, A Biographical Memoir*. National Academies Press, Washington DC, USA, 1995.
- [Sh03] Shaffer, Dale; Doube, Wendy; Tuovinen, Juhani; Sturt, Charles; Wagga, Wagga; Wales, New South: In (Petre, M.; Budgen, D., Hrsg.): *Proceedings of the Joint Conference EASE & PPIG 2003*. 2003.
- [SS11] Schubert, Sigrid; Schwill, Andreas: *Didaktik der Informatik*. Spektrum Akademischer Verlag, Heidelberg, 2. Auflage, 2011.
- [Wi91] Wirth, Niklaus: *Die Dinge beim Namen nennen*. In: *Erstes European Software Festival*. Chip – Vogel-Verlag, Würzburg, S. 60–64, 1991.
- [Wi03] Witten, Helmut: *Allgemeinbildender Informatikunterricht? Ein neuer Blick auf H. W. Heymanns Aufgaben allgemeinbildender Schulen*. INFOS 2003, GI, Bonn, S. 59–75, 2003.
- [Wi06] Wing, Jeannette M.: *Computational Thinking*. *Commun. of the ACM*, 49(3):33–35, 2006.
- [Wi08] Wing, Jeannette M.: *Computational thinking and thinking about computing*. *Philosophical Transactions of The Royal Society*, (366):3717–3725, 2008.

Von Eingebetteten Systemen zu Physical Computing: Grundlagen für Informatikunterricht in der digitalen Welt

Mareen Przybylla¹ Ralf Romeike²

Abstract: Der digitale Wandel basiert neben der zunehmenden Durchdringung des täglichen Lebens mit digitalen Systemen auch auf der parallel stattfindenden Entwicklung von untereinander vernetzten allgegenwärtigen, sogenannten cyber-physischen Systemen. Diese sind in der Lage, mit Sensorik ihre Umwelt zu erfassen und mittels Aktorik in die menschliche Erfahrungswelt zurückzuwirken. Für die Schulinformatik wird das Potenzial dieser Innovationen häufig noch unzureichend umgesetzt, was nicht zuletzt in Ermangelung eines geeigneten fachlichen Rahmens und daraus resultierend fehlender Lehr-Lern-Materialien begründet ist. Dieser Beitrag untersucht daher die fachlichen Grundlagen des Themenkomplexes und identifiziert für die Schulinformatik relevante Inhalte und stellt sie strukturiert dar. Anhand einer interaktiven Modellstadt wird exemplarisch gezeigt, wie die Inhalte im Informatikunterricht verankert werden können.

Keywords: Eingebettete Systeme; Physical Computing; Fachstruktur; Unterrichtsbeispiel

1 Einleitung

Immer weniger technische Entwicklungen der heutigen Zeit nutzen klassische Computer mit Bildschirm, Tastatur und Maus. Wir sind vorwiegend von mobilen Geräten und reaktiven Systemen umgeben, die anhand von Sensoren und Aktoren kontinuierlich mit ihrer Umwelt kommunizieren. Mikrocontroller dominieren in Gestalt von eingebetteten Systemen unseren Alltag. Im Schulunterricht werden hingegen meist klassische Desktop-PCs als Beispiel herangezogen, die nur einen Bruchteil aller Informatiksysteme ausmachen. Der Umgang mit eingebetteten Systemen erfordert neues Wissen und neue Kompetenzen, die im traditionellen Unterricht nicht erworben werden. Damit Schüler am gesellschaftlichen Diskurs teilhaben, der Medienberichterstattung folgen und informiert Urteile fällen können, ist es notwendig, die relevanten Aspekte im Unterricht zu thematisieren. Um ihnen darüber hinaus einen produktiven und kreativ-gestalterischen Umgang mit modernen Informatiksystemen zu ermöglichen, muss Unterricht sich an den Gegebenheiten der realen Welt orientieren und entsprechende Kompetenzen ausbilden. Zwar gibt es bereits zahlreiche Beispiele zur Verwendung von Mikrocontrollern im Unterricht, jedoch werden diese häufig rein als Werkzeug und nicht als Unterrichtsgegenstand thematisiert. Hierzu fehlt bisher ein geeigneter fachlicher Rahmen. Dieser Beitrag untersucht daher Literatur der Fachwissenschaft mit

¹ Universität Potsdam, Didaktik der Informatik, Augut-Bebel-Straße 89, 14482 Potsdam przybyll@uni-potsdam.de

² Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstraße 3, 91058 Erlangen, ralf.romeike@fau.de

dem Ziel, für den Informatikunterricht relevante Inhalte zu extrahieren und geeignet zu strukturieren. Die Ergebnisse können dann genutzt werden, um in diesem Bereich angesiedelten Unterricht zu strukturieren, geeignete Unterrichtsmethoden zu etablieren oder Lern- und Lehrmaterialien zu erstellen, die auf die entsprechende Zielgruppe zugeschnitten sind. Anhand einzelner Beispiele aus einem Projekt zur Gestaltung einer interaktiven Modellstadt wird schließlich gezeigt, wie sich ausgewählte Probleme dieses Fachbereichs im Rahmen von Physical Computing im Informatikunterricht thematisieren lassen.

2 Eingebettete Systeme in der informatischen Schulbildung

Unter anderem durch die immer stärkere Präsenz unzähliger Mikrocontroller in autonomen Fahrzeugen, „Smart“-Entwicklungen und Alltagsgegenständen sind eingebettete Systeme seit einiger Zeit einer der wichtigsten und innovativsten Forschungs- und Entwicklungsschwerpunkte der Informatik. Auch die Schulinformatik beschäftigt sich grundsätzlich seit langem mit diesem Themenbereich. Baumann zeigte beispielsweise in einem historischen Abriss zu Entwicklungen in der Schulinformatik im Forschungsgebiet *Eingebettete Systeme*, dass seit etwa 30 Jahren regelmäßig Artikel zu den Themen *Messen – Steuern – Regeln*, *Prozessdatenverarbeitung* und *Automatisierung* veröffentlicht wurden [Ba11]. Im kürzlich erschienenen LOGIN-Themenheft „Eingebettete Systeme“ [Fa16] werden zahlreiche Unterrichtsbeispiele und verschiedenste Ansätze und Werkzeuge für den Unterricht vorgestellt. Aus fachdidaktischer Sicht analysiert z. B. Stechert ausführlich den Informatiksystembegriff und schlussfolgert, dass es zur Kompetenzentwicklung in diesem Bereich nötig sei, „Informatiksysteme in ihrer Einheit aus Hardware, Software und Vernetzung anhand ihres nach außen sichtbaren Verhaltens, der inneren Struktur und Implementierungsaspekten zu analysieren“ und typische Repräsentanten von Informatiksystemen mit Schülern zu untersuchen und sie zu erklären [St09]. In den GI-Bildungsstandards Informatik für die Sekundarstufe II wird empfohlen, zur Entwicklung von Informatiksystemen maschinell verarbeitbare Sachverhalte der realen Welt zu identifizieren und modellieren. Als typische Einsatzbereiche werden u. a. *Robotik*, *Prozesssteuerung* und *Prozessregelung* genannt [Ar16]. Auch in den neueren Curricula einiger Bundesländer wird diese Entwicklung ersichtlich. So empfiehlt der Berlin-Brandenburger Rahmenlehrplan beispielsweise die Realisierung eines eigenen *Physical-Computing-Projektes* und die Verwendung externer Hardware innerhalb des Themenfeldes *Informatiksysteme* [BM15]. Im Wahlthemenfeld Physical Computing wird die *Charakterisierung eingebetteter Systeme*, die *Verarbeitung elektrischer Größen* oder auch die *Verwendung von Mikrocontrollern* in verschiedenen Kontexten verlangt [BM15]. Lehrkräfte zeigen immer wieder reges Interesse an Fortbildungen, Bausätzen und Unterrichtsmaterialien. Verfügbare Ressourcen beschränken sich jedoch häufig auf die Verwendung von Werkzeugen zur motivierenderen Gestaltung des traditionellen Unterrichts und gehen nur selten über die Grundlagen der Sensor-Aktor-Ansteuerung hinaus (z. B. [Ba11; NW16; PR12]), sodass Phänomene des Ubiquitous Computing mit den bisherigen Ansätzen im Unterricht nicht ausreichend erklärt werden.

3 Phänomene der digitalen Welt: Ubiquitous Computing

Informatiksysteme entwickelten sich in den letzten Jahrzehnten immer mehr entsprechend der Vision des *Ubiquitous-Computing*-Pioniers Mark Weiser, der bereits Anfang der 1990er Jahre allgegenwärtige, effiziente und gleichzeitig unsichtbare Computer vorhersagte [We93]. Diese Vision ist durch entsprechende Technologien heute alltäglich und wird durch das *Internet der Dinge* (engl. Internet of Things, IoT) und *Internet von Allem* (engl. Internet of Everything, IoE) noch ausgeweitet. Im IoT werden „Dinge“, die *eingebettete Systeme* enthalten, untereinander und mit Diensten im Internet vernetzt. Geräte und Maschinen erfassen mit Sensorik ihre Kontexte und übermitteln diese Daten. Insbesondere durch den Zusammenfluss der Daten vieler Dinge entstehen neue Möglichkeiten der Informationsgewinnung, die physikalische Welt verschmilzt mit der virtuellen. Für die Industrie ist dies vor allem im Bereich intelligenter Maschinen von Bedeutung, aber auch in der Heimanwendung finden sich Beispiele: Autos, Haushaltsgeräte oder beliebige Einrichtungsgegenstände können mit Diensten aus dem Internet verknüpft werden. Die Daten werden beispielsweise zur Fahrzeugnavigation, automatisierten Ersatzteilbestellung oder Marktanalyse genutzt. Die Einbindung solcher Dienste ermöglicht schließlich dank Echtzeitanalysen, Einfluss auf die reale Welt zu nehmen. Durch das stetige Anwachsen der Informationsvielfalt im Internet und über das Vernetzen der Dinge mit Personen und Prozessen, wird aus dem IoT immer mehr das IoE [BHV14]. Um hierzu inhaltlich fundierten Unterricht anbieten zu können, bedarf es einer fachlichen Klärung unter didaktischen Gesichtspunkten. Die Sichtung von Lehr- und Lernmaterialien zu den genannten Themen zeigte, dass es eine große Fülle an Lehrbüchern der Fachwissenschaft gibt, die jedoch für eine gänzlich andere Zielgruppe konzipiert worden sind (oft Studierende der Ingenieurwissenschaften und Informatik, Hard- und Software-Entwickler in der Praxis). Diese können nicht unmittelbar für den allgemeinbildenden, wissenschaftspropädeutischen Einsatz im Schulunterricht genutzt werden. Es stellt sich die Frage, wie die Fachstruktur zu beschreiben ist, sodass ihre zentralen Inhalte im Unterricht adäquat vermittelt werden können. Daher wurden vorhandene Lehrbücher mit dem Ziel analysiert, die Grundlagen herauszuarbeiten und Inhalte für die Gestaltung entsprechender Unterrichtseinheiten im Informatikunterricht zu extrahieren und geeignet zu strukturieren. Hierzu wurde in einem ersten Analyseschritt aus der Literatur zunächst zusammengetragen, welche informatischen Inhalte zu den jeweiligen Fachgebieten allgemein thematisiert werden. Im zweiten Schritt wurde der daraus entstandene Katalog hinsichtlich zentraler Aspekte analysiert, also solcher Inhalte, Methoden und Konzepte, die in allen Werken als charakteristische Merkmale des betreffenden Gebietes beschrieben und für das Verständnis erforderlich sind. Diese wurden auf ihre wesentlichen Eigenschaften reduziert. Im letzten Schritt wurden Gemeinsamkeiten zusammengefasst. In den folgenden Abschnitten werden die einzelnen Forschungsgebiete näher charakterisiert.

3.1 Eingebettete Systeme

Eingebettete Systeme (ES) vereinen Hard- und Softwarekomponenten, werden im Gegensatz zu Mehrzweckcomputern in größere Produkte eingebettet und erfüllen in der Regel genau

einen Zweck. Sie sind meist mit physischen Prozessen in ihrer Umgebung durch Sensoren und Aktoren verbunden und erfordern oft keine menschlichen Eingaben (vgl. [VG02]). Beispiele für ES finden sich in allen denkbaren Bereichen unseres Lebens: im Straßen- und Bahnverkehr, in medizinischen Geräten oder im „Smart Home“. Der Verband BITKOM definiert ES insbesondere unter Einbezug ihres typischen Zwecks, Systeme zu steuern, regeln oder überwachen. ES erfüllen vordefinierte Aufgaben, oft unter Echtzeit-Anforderungen [BI10]. Sensor- und Aktor-Schnittstellen ermöglichen es, Veränderungen in der Umwelt zu erfassen und Änderungen an der Umwelt vorzunehmen. Im Gegensatz zu transformationellen Systemen, die Eingabe, Verarbeitung und Ausgabe linear durchlaufen und dann ihr Programm beenden, laufen ES kontinuierlich und interagieren stetig mit ihrer Umgebung. Typische Hardware für den Informatikunterricht stellen Mikrocontroller, wie z. B. Arduino dar.

3.2 Robotik

Ein spezielles Anwendungsgebiet von ES ist die Robotik. Roboter sind insbesondere autonom agierende Maschinen, die in einen Bewegungsapparat eingebunden sind und beispielsweise Arme, Beine oder Räder zur Fortbewegung oder für Greiftätigkeiten nutzen. Sie unterstützen und ersetzen Menschen durch physische Aktivität beim Ausführen konkreter Aufgaben und können ihre Umwelt modifizieren [Si10]. Hertzberg et al. betonen zudem die mit der Autonomie von mobilen Robotern verbundene Schwierigkeit, vorhersehbare Programme zu erstellen. Dadurch, dass all ihre Aktionen von ihrer Umgebung abhängig sind, die erst zur Ausführungszeit bekannt ist, müssen Roboter diese mit Sensoren erfassen, die Daten bewerten und passende Handlungsweisen initiieren [HLN12]. Robotik befasst sich eingehend mit Themen wie Kinetik, mechanischen Bewegungsabläufen, Wahrnehmung, Navigation und Hindernisvermeidung. Roboter können auch dort eingesetzt werden, wo es für Menschen schwierig ist zu arbeiten, z. B. im All, in der Wüste oder im Meer. Sie werden zur Effizienzsteigerung in der Lebensmittel- und Güterherstellung, Land- und Forstwirtschaft oder Medizintechnik verwendet. Immer häufiger finden sie sich beispielsweise in Form von Staubsauger- oder Bodenwischrobotern auch in Haushalten wieder. Unterrichtsbeispiele mit LEGO Mindstorms sind überwiegend der Robotik zuzuordnen.

3.3 Cyber-Physische Systeme

Cyber-physische Systeme (CPS) haben viele Gemeinsamkeiten mit ES und werden in der Literatur oft synonym behandelt. In CPS überwachen und steuern untereinander und mit dem Internet vernetzte ES physische Prozesse. Dies geschieht üblicherweise mit Regelungstechnik, indem Daten aus der Umwelt die Berechnungsprozesse beeinflussen und umgekehrt. Neben der physischen Umgebung (z. B. mechanische Teile, chemische Prozesse, menschliche Bediener, also die direkte Umgebung des Systems) bestehen CPS aus Rechnerplattformen, die neben den Mikrocontrollern auch Sensorik und Aktorik bereitstellen und Netzwerkkomponenten, die die Kommunikation der Teilsysteme untereinander und über das Internet ermöglichen [LS14].

3.4 Gemeinsamkeiten von ES, Robotern und CPS

Obwohl die beschriebenen Systemarten jeweils in verschiedene Richtungen spezialisiert sind, gibt es eine Reihe an Eigenschaften und Anforderungen, die übergeordnet als zentral angesehen werden können und aus denen sich zentrale Konzepte und Ideen ableiten lassen. Grundsätzlich werden in allen Systemen *Sensoren* verwendet, um Eingangssignale auszuwerten, zu verarbeiten und entsprechend der darauf basierenden und durch Software bestimmten Entscheidungen *Aktoren* angesteuert, um etwas zu kontrollieren, zu steuern, zu bewegen – also ein Signal in die Außenwelt zu geben. Man unterscheidet hinsichtlich der Datenerfassung allgemein zwischen *zeitkontinuierlichen Systemen*, die, meist als Regelungssysteme, ununterbrochen Signalströme verarbeiten und *diskreten Systemen*, die ereignis- oder zeitgesteuert diskrete Signale verarbeiten. Je nach Anwendungsbereich müssen ES, Roboter und CPS strengen Anforderungen genügen, wie beispielsweise *Echtzeit-Anforderungen*, innerhalb eines vordefinierten Zeitraums korrekte Ergebnisse zu liefern [VG02, pp. 69-74]. Dies ist vor allem in sicherheitskritischen Bereichen von Bedeutung, wie in automatisierten Zugsystemen oder in der computergestützten Chirurgie. Bei Echtzeitsystemen wird zwischen *weicher*, *moderater* und *harter* Echtzeit unterschieden, also danach, ob ein Überschreiten einer Zeitschranke als Systemfehler gewertet wird oder in gewissem Maß tolerabel ist. Während im ersten Fall ein Versagen katastrophale Folgen haben kann, beeinflusst es in letzterem lediglich die Qualität des Systems. Die Laufzeit kann daher kritisch werden und Einfluss auf die Korrektheit nehmen, statt nur auf Performanz oder Bedienkomfort [LS14]. Die meisten ES und CPS lassen die Ausgabe eines physischen Systems einer gegebenen Referenzeingabe folgen (z. B. Thermostat). Im deutschsprachigen Raum werden hierbei weiter Steuern und Regeln unterschieden: Im Steuersystem wird z. B. das Heizungsventil anhand der Temperatureingabe 22°C entsprechend der Berechnung für die Referenzeingabe geöffnet. In Regelungssystemen wird entsprechend der gemessenen Realtemperatur nachreguliert, sollte die erreichte Temperatur nicht 22°C betragen. Solche Systeme sollen sich sinnvoll verhalten und die Referenzeingabe auch bei Rauschen, Modellfehlern oder Störungen verfolgen. Auch hier gibt es wieder spezielle Anforderungen (vgl. [LS14; VG02]):

- *Stabilität*: Ausgangssignale sollen bei gleichbleibenden Eingangssignalen ihre Werte möglichst nicht verändern
- *Performanz*: gewünschte Ausgabe wird möglichst exakt in möglichst kurzer Zeit an Referenzeingabe angepasst
- *Störungsabwehr*: Minimaler Einfluss von Störungen auf das System
- *Robustheit*: Minimaler Einfluss von Modellfehlern auf Stabilität und Performanz

In aller Regel unterliegen ES, Roboter und CPS in der Entwicklung strengen Vorgaben (sogenannten *Entwurfsmetriken*) bezüglich *Kosten*, *Stromverbrauch*, *Formfaktor* und *Performanz*. Auch die Entwicklungszeit spielt in der Wirtschaft eine große Rolle. Unterschieden wird hierbei zwischen *Time-to-Prototype (TTP)* und *Time-to-Market (TTM)*, die häufig ausschlaggebend für den Erfolg oder Misserfolg eines Produktes sind. Alle genannten

Metriken stehen gegenseitig in Konkurrenz [VG02]. So hat erhöhte Performanz z. B. oft auch einen höheren Stromverbrauch und damit sinkende Energieeffizienz zur Folge. Der Entwurf von ES, CPS und Robotern bringt eine Reihe an Problemstellungen mit sich, aus denen sich typische Methoden und Konzepte zur Problemlösung ableiten lassen. Dazu gehören u. a. *Nebenläufigkeit*, *Trade-Offs* und *Heterogenität*:

- *Nebenläufigkeit*: Prozesse in der realen, physischen Welt finden parallel statt, müssen aber in einer sequentiellen Semantik abgebildet werden. Daher werden Konzepte wie *Interrupts*, *Multitasking* oder *Semaphore* genutzt, um beispielsweise *Nachrichtenaustausch* zu ermöglichen und *Deadlocks* zu vermeiden [LS14].
- *Trade-Offs*: Es muss bei der Gestaltung solcher Systeme abgewogen werden zwischen schnell berechneten, aber weniger genauen und langsamer berechneten, aber akkuraten Ergebnissen [Wo09].
- *Heterogenität*: ES, Roboter und insbesondere CPS sind schwieriger zu analysieren und entwerfen als homogene Systeme, da sie neben der Kombination aus Netzwerk-kommunikation und der eigentlichen Berechnung zusätzlich dynamische, physische Prozesse einbeziehen, die nie vollumfänglich vorhersehbar sind [LS14].

3.5 Zusammenfassung der Fachstruktur

Es zeigt sich, dass die beschriebenen Forschungsgebiete stark verzahnt sind und dass das Verständnis bestimmter Inhalte und Konzepte Voraussetzung ist, um andere Gebiete zu erschließen. Da sie die fachliche Grundlage für das gesamte Fachgebiet bilden, sollten die vorgestellten Eigenschaften, Anforderungen und Problemstellungen sowie daraus resultierende Methoden und Konzepte eingebetteter Systeme auch im Informatikunterricht thematisiert werden. Die in Abb. 1 dargestellte Struktur des Fachgebietes verortet die einzelnen Forschungsgebiete im größeren Zusammenhang und stellt ihre Unterscheidungsmerkmale dar. Neben den oben beschriebenen Bereichen gibt es noch zahlreiche weitere Technologien im Kontext der *Mensch-Maschine-Interaktion*, beispielsweise *Interaktive Systeme*, *Wearable Computer* oder auch *Elektronische Textilien*, die sich aus weniger technischen Perspektiven mit dem Gegenstandsbereich eingebetteter Systeme auseinandersetzen.

4 Fachdidaktische Aufarbeitung

Die vorgestellte Struktur kann herangezogen werden, um notwendige Vorkenntnisse der Schüler und Zusammenhänge zwischen den Teilgebieten herstellen und sichern zu können. Zentrale Inhalte und Konzepte eingebetteter Systeme werden zur Entwicklung verschiedenster interaktiver Systeme benötigt, sodass je nach Zielgruppe und Interessen unterschiedliche Themengebiete mit verschiedenen Schwerpunkten als Kontexte im Unterricht genutzt werden können. Auf diese Weise können Kompetenzen zur Analyse, Modellierung und Umsetzung von ES schrittweise entwickelt werden. Das Gestalten eingebetteter Systeme stellt zunächst

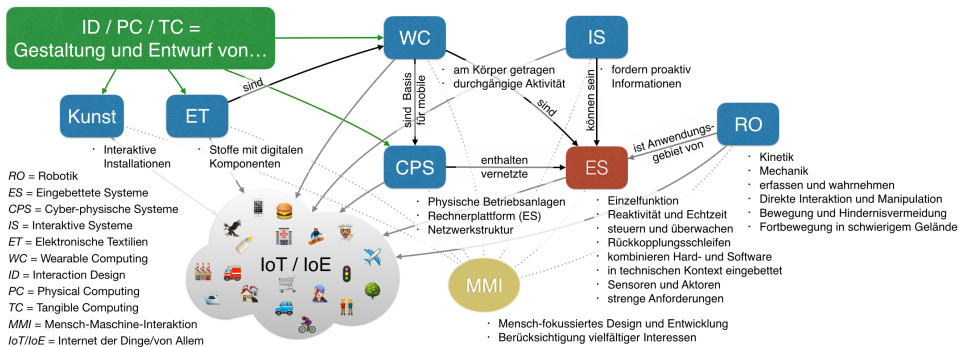


Abb. 1: Überblick über die verschiedenen Forschungsgebiete im Bereich eingebetteter Systeme

eine große Herausforderung dar, die die Beachtung und Bearbeitung zahlreicher Teilaspekte beinhaltet. Daher ist es wichtig, den Lernenden neben der inhaltlichen Auseinandersetzung mit dem Thema auch Werkzeuge und Methoden zur Verfügung zu stellen, die Arbeitsschritte strukturieren und so erfolgreiches Lernen ermöglichen. Eine Möglichkeit hierzu bietet *Physical Computing*: Programmierbare Mikrocontroller bieten als Werkzeug eine attraktive und vielversprechende Heranführung an die Gestaltung eingebetteter Systeme.

4.1 Umsetzung mit Physical Computing

Physical Computing hat sich mit einer stetig wachsenden Gemeinschaft aus Künstlern, Designern und Hobby-Bastlern entwickelt, die Mikrocontroller nutzen, um interaktive Objekte herzustellen. Hierfür werden Sensoren, wie Lautstärke- oder Temperatursensoren und Aktoren, wie Servomotoren oder Lautsprecher verwendet, um kontinuierlich mit ihrer Umwelt zu interagieren. Je nach Ziel des jeweiligen Projektes, werden bestimmte Problemstellungen, Anforderungen und Entwurfsmetriken aus den in Abschnitt 3 analysierten Fachgebieten relevant. Zusammenfassend betrachten wir Physical Computing als die *kreative Gestaltung und Entwicklung interaktiver, physischer Objekte und Systeme*, die als programmierte, greifbare Medien über *Sensoren und Aktoren* mit ihrer Umwelt kommunizieren [PR12]. Physical Computing nutzt Methoden und Ideen eingebetteter Systeme, derer Spezialgebiete und Anwendungen. Es überträgt die traditionell virtuellen kreativen Möglichkeiten der Informatik durch Einbezug von Aspekten aus Kunst und Design in die reale Welt und kann das Fach vielfältiger und damit attraktiver erscheinen lassen. Auch der hohe Motivationswert selbst kreierter Produkte lässt sich für den Informatikunterricht positiv nutzen.

4.2 Beispielumsetzung: Projekttag Smart City

Die nachfolgenden Beispiele von einem Projekttag, der mit 22 Schülerinnen und Schülern der achten bis elften Jahrgangsstufen durchgeführt wurde, illustrieren exemplarisch, wie

Inhalte aus den Bereichen ES, CPS und Robotik in Informatikprojekten verankert werden können. Ziel des Projektes war der Bau einer interaktiven LEGO-Stadt („Smart City“). In einer solchen Stadt gibt es zahlreiche eingebettete Systeme. Diese erfassen an verschiedenen Stellen ihre Umwelt (z. B. Wettereinflüsse, Verkehrsaufkommen, Müllstand) und steuern diese (z. B. Markisen einfahren, Ampelsteuerung anpassen, Müllabholung regulieren).

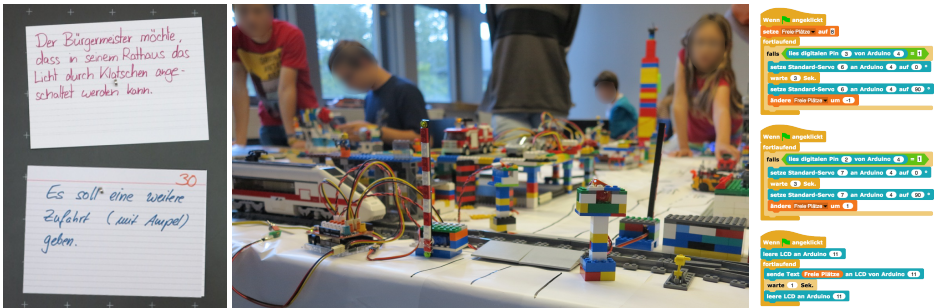


Abb. 2: Schüler beim Bau einer „Smart City“ mit LEGO, Arduino TinkerKit und Snap4Arduino

In kleinen Projektgruppen arbeiteten die Teilnehmer jeweils an Teilprojekten einer interaktiven LEGO-Stadt (siehe Abb. 2). Alle Schüler kamen unabhängig von ihrer konkreten Aufgabe mit gewissen Grundkonzepten eingebetteter Systeme in Berührung, wie beispielsweise der *Kalibrierung* und *Ansteuerung* von *Sensoren* und *Aktoren* oder *zeitkontinuierlichen* und *diskreten* Signalen. Darüber hinaus lernten sie an konkreten, nicht konstruierten Beispielen typische Probleme des Entwurfs eingebetteter Systeme kennen und mit ihnen umzugehen. Durch das Projektziel, die „Smart City“ noch am selben Tag zu präsentieren, wurde eine Situation geschaffen, in der die *Time-to-Market*, wie auch in industriellen Projekten, unausweichlich festgelegt war. Das Projekt wurde mit agilen Methoden umgesetzt, sodass auch die *Time-to-Prototype* begrenzt war, indem die Schüler nach vorab definierten Zeiträumen ihre Prototypen den „Bürgermeistern“ (Betreuern) vorstellen mussten.

Unsere früheren Erfahrungen zeigten, dass es für Schüler häufig eine besondere Herausforderung darstellt, physikalische Prozesse der Umgebung dem Ziel entsprechend zu erfassen. Dies lässt sich mit Physical Computing gut bewerkstelligen, indem sich Unterricht an Leitfragen von O’Sullivan und Igoe [OI04] orientiert, z. B. Was soll das Produkt aus Sicht des Benutzers tun? (*Planung aus Nutzersicht*), Was sind Eingaben, Verarbeitungsschritte und Ausgabe des Produktes aus nicht-technischer Sicht? Welche Ein- und Ausgaben sind diskret, welche zeitkontinuierlich? (*Planung aus Entwicklersicht*). Im Projekt umfasste die Planung aus Nutzersicht eine Beschreibung der Ziele derart, dass sie für die „Bürgermeister“ der Stadt verständlich sein sollte, da von ihnen keinerlei Technikverständnis vorausgesetzt werden durfte. Auch die Bürgermeister selbst konnten Wünsche äußern und an das Projektboard heften. Die Planung aus Entwicklersicht wurde gruppenintern in der Kürze der Zeit nur rudimentär umgesetzt. Die Betreuer regten in den Gruppen entsprechende Diskussionen an.

Während harte Echtzeit in Schulprojekten meist keine Rolle spielen wird, sind *weiche*

Echtzeitanforderungen an die zu erstellenden ES häufig zu erfüllen. Typische Probleme tauchen immer dann auf, wenn Verzögerungen für die Nutzer wahrnehmbar werden. Insbesondere bei gezielten Nutzerinteraktionen ist dies der Fall. In einem Teilprojekt hatten Schüler beispielsweise das Problem zu lösen, dass die Rathausbeleuchtung in der Stadt drahtlos aktiviert werden sollte, jedoch die Übertragungsverzögerung über WLAN deutlich spürbar war. Die Vorhersehbarkeit von Umwelteinflüssen ist ein typisches Problem *heterogener Systeme*. In der Stadt sollten zum Beispiel Fahrzeuge gezählt werden, die einen Parkplatz befahren. Bei der Verwendung eines Helligkeitssensors in der Bodenplatte wurde schnell deutlich, dass Umwelteinflüsse die Zählung verfälschen können. Die Schüler mussten daher eine Lösung für das Problem finden, dass Schwankungen in der Helligkeit über dem Sensor nicht immer bedeuteten, dass ein Fahrzeug die Stelle überfuhr. In Bezug auf Zuverlässigkeit ist *Störungsabwehr* einer der zentralen Aspekte neben Robustheit im Entwurf eingebetteter Systeme. So sollte in der Stadt zum Beispiel ein beschränkter Bahnübergang zuverlässig sensorgesteuert schließen und nach Zugdurchfahrt wieder öffnen. Dabei war es besonders wichtig, keine Fehler in der Durchfahrterkennung zuzulassen.

5 Diskussion

Allgemeinbildender Informatikunterricht steht vor der Herausforderung, zeitbeständige Ideen und Konzepte zu vermitteln und Kompetenzen zu schulen, die in der digitalen Welt von Bedeutung sind. Einige Innovationen der letzten Jahrzehnte und damit zusammenhängende Ideen und Konzepte sind bisher im Schulunterricht jedoch nicht ausreichend verankert: Moderne Informatiksysteme sind immer häufiger eingebettete Systeme, die hochgradig vernetzt und überall zu jeder Zeit verfügbar sind. In diesem Beitrag wurde begründet, dass Themen aus den Bereichen eingebettete Systeme, cyber-physische Systeme und Robotik mit all den dazugehörigen Konzepten (wie z. B. Steuern und Regeln, Nebenläufigkeit oder Trade-Offs) im Informatikunterricht geeignet sind, Schüler dazu zu befähigen, produktiv und kreativ solche Systeme zu gestalten und kompetent mit ihnen umzugehen. Anhand der Erkenntnisse aus den Forschungsgebieten wurde deutlich, dass es zahlreiche gemeinsame Merkmale gibt, die für den Unterricht relevant sind und somit eine fachlich-inhaltliche Grundlage liefern. Die dargestellte Struktur des Fachgebietes kann genutzt werden, um eine inhaltlich logisch aufgebaute Unterrichtsplanung vorzunehmen, Teilgebiete im größeren Zusammenhang zu verorten und geeignete Lehr-Lernszenarien und -materialien für die Schule zu entwerfen. Mit den zugrundeliegenden Konzepten lassen sich auf sehr motivierende und kreative Weise interaktive Objekte und Systeme gestalten, wie z. B. Arbeiten aus der Maker-Kultur zeigen. Es wurde am Beispiel einer Modellstadt aufgezeigt, wie solche Inhalte im Rahmen von Physical Computing kontextualisiert werden können. Unsere Unterrichtserfahrungen zeigen, dass Physical Computing eine attraktive Herangehensweise ist, sich den Inhalten nicht nur auf theoretischer Ebene zu nähern, sondern mit geeigneten Werkzeugen auch praktisch umzusetzen. Beim Entwurf interaktiver, physischer Objekte und Systeme werden Interaktionen zwischen Teilsystemen mit allen dazugehörigen Problemen thematisiert. Diese Inhalte werden mit Physical Computing für den Informatikunterricht auf einer sehr konkreten Ebene zugänglich gemacht.

Literatur

- [Ar16] Arbeitskreis »Bildungsstandards SII« der Gesellschaft für Informatik e.V.: Bildungsstandards Informatik für die Sekundarstufe II, 2016.
- [Ba11] Baumann, R.: Eingebettete Systeme verstehen. Teil 1: Kreatives Experimentieren mit Arduino. LOG IN 171/, S. 33–45, 2011.
- [BHV14] Bojanova, I.; Hurlburt, G.; Voas, J.: Imagineering an Internet of Anything. Computer 47/6, S. 72–77, 2014.
- [BI10] BITKOM: Eingebettete Systeme – Ein strategisches Wachstumsfeld für Deutschland, 2010.
- [BM15] Berliner Senatsverwaltung für Bildung Jugend und Familie; Ministerium für Bildung Jugend und Sport des Landes Brandenburg: Teil C Informatik, 2015.
- [Fa16] Eingebettete Systeme. LOG IN - Informatische Bildung und Computer in der Schule/185/186, hrsg. von Fachbereich Erziehungswissenschaft und Psychologie der Freien Universität Berlin, 2016.
- [HLN12] Hertzberg, J.; Lingemann, K.; Nüchter, A.: Mobile Roboter - Eine Einführung aus Sicht der Informatik. Springer-Verlag, Berlin Heidelberg, 2012.
- [LS14] Lee, E. A.; Seshia, S. A.: Introduction to Embedded Systems - A Cyber-Physical Systems Approach. 2014.
- [NW16] Neutens, T.; Wyffels, F.: Teacher professional development through a physical computing workshop. In: Proceedings of the 11th Workshop in Primary and Secondary Computing Education (WiPSCE '16). S. 108–109, 2016.
- [OI04] O'Sullivan, D.; Igoe, T.: Physical Computing: Sensing and Controlling the Physical World with Computers. Thomson Course Technology, Boston, 2004.
- [PR12] Przybylla, M.; Romeike, R.: My Interactive Garden - A Constructionist Approach to Creative Learning with Interactive Installations in Computing Education. In: Proceedings of Constructionism 2012. Athen, S. 395–404, 2012.
- [Si10] Siciliano, B.; Sciavico, L.; Villani, L.; Oriolo, J.: Robotics: Modeling, Planning and Control. 2010.
- [St09] Stechert, P.: Fachdidaktische Diskussion von Informatiksystemen und der Kompetenzentwicklung im Informatikunterricht, Diss., Universität Siegen, 2009, 45ff., 309.
- [VG02] Vahid, F.; Givargis, T. D.: Embedded System Design: A Unified Hardware/Software Introduction. Wiley, 2002.
- [We93] Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. Communications of the ACM 36/7, S. 75–84, 1993.
- [Wo09] Wolf, W.: Cyber-physical systems. IEEE Computer 42/3, S. 88–89, 2009.

Quadrologik – Modellbildung und Modularisierung auf Basis von Rechner-technik

Tobias Jördens¹, Jens Gallenbacher²

Abstract: Auf der Abstraktionsebene oberhalb von Transistoren sind für den Computere-ntwurf Logikgatter wichtige Grundbausteine. Die Behandlung von Logikgattern im Unterricht gehört aber nicht nur zu einer vertieften Beschäftigung mit dem Aufbau von Informatiksystemen. Anhand des Entwurfs mit Logikgattern kann man Kompetenzentwicklung in Modellbildung fördern und die fundamentale Idee der Modularisierung adressieren. Eine Vorentlastung oder Rückbindung an die notwendige Einführung boolescher Ausdrücke im Rahmen der Einführung der algorithmischen Grundbausteine ist möglich. Für die Beschäftigung von Logikgattern im Unterricht entwickelten wir ein neues Logikgatterstecksystem, welches in diesem Artikel vorgestellt wird und dessen erste Einsätze im Unterricht erläutert werden. Schwerpunkt sind hier Unterrichtsstunden mit Schülerinnen und Schülern der Jahrgangsstufen 4 und 5.

Keywords: Modularisierung, Modellbildung, Technische Informatik, Digitaltechnik, Logikgatter, Logikstecksystem, Logische Schaltungen, Rechner-technik

1 Einleitung

Andreas Schwill hat in seiner Erläuterung der fundamentalen Ideen der Informatik [Sc93] als eine von drei Masterideen die strukturierte Zerlegung begründet und diese aufgefächert in Hierarchisierung, Modularisierung und Orthogonalisierung. Wie kann man diese im Unterricht adressieren? Unser Ansatz ist es, insbesondere Modularisierung im Kontext von Rechner-technik zu vermitteln.

Auch die wichtige Kompetenz der Modellierung kann darüber gefördert werden. Logikgatter auf Papier, in Simulatoren oder in Logikstecksystemen sind Modelle realer elektronischer Bauteile. Die Erstellung von Schaltungen aufgrund einer Problemstellung ist an sich bereits ein Modellierungsprozess. Geht man auf höhere Abstraktionsebenen des Rechnerentwurfs, so wird dieser Aspekt noch verstärkt.

Warum wir Rechner-technik als Unterrichtsgegenstand zur Vermittlung von Modularisierung und Modellierung als sinnvoll erachten, erläutern wir in Kapitel 2 dieses Beitrags. Für die Frage nach dem *wie* des Unterrichts ist für uns die Beachtung der enaktiven Ebene wichtig. Eine kurze Vorstellung und Bewertung bisher vorhandener Möglichkeiten erfolgt daher in Kapitel 3, bevor dort das aktuell bei uns in der

¹ Schuldorf Bergstraße und Technische Universität Darmstadt, Didaktik der Informatik, Hochschulstraße 10, 64289 Darmstadt, tj@di.tu-darmstadt.de

² Technische Universität Darmstadt, Didaktik der Informatik, Hochschulstraße 10, 64289 Darmstadt, jg@di.tu-darmstadt.de

Entwicklung befindliche System Quadrologik vorgestellt und erläutert wird. Die bisherigen Unterrichtsversuche mit diesem System werden wir im Kapitel 4 vorstellen, bevor in Kapitel 5 der Ausblick gegeben wird.

2 Motivation und Legitimation für Rechnertechnik als Unterrichtsthema

Computer sind heute allgegenwärtig. Für Schülerinnen und Schüler ist es selbstverständlich, dass sie sich nicht nur für die in Software realisierten Konzepte der Informatik interessieren, sondern auch für die Wirk- und Gestaltungsprinzipien der darunterliegenden Informatiksysteme.

Wenngleich die Motivation für die Schülerinnen und Schüler also einfach geklärt ist, so existieren für die Frage der Behandlung spezifischer Inhalte im Informatikunterricht in der Literatur verschiedene Legitimationsgrundlagen. In den Bildungsstandards Informatik [GI08] wird für den Inhaltsbereich Informatiksysteme als Kompetenz formuliert: „*Schülerinnen und Schüler aller Jahrgangsstufen verstehen die Grundlagen des Aufbaus von Informatiksystemen und deren Funktionsweise*“ [ebd. S.37]. Bei der Präzisierung dieser Kompetenz verbleibt die Behandlung jedoch, fast ausschließlich, auf der Ebene der mit dem bloßen Auge sichtbaren Hardwarekomponenten [vgl. ebd.]. In Lehrplänen³ findet man allerdings auch die Behandlung von Abstraktionsebenen unterhalb der sichtbaren Hardware. Entsprechend gibt es auch Unterrichtseinheiten zu diesem Thema. So stellt Urs Lautenbach eine Unterrichtseinheit vor, die „Vom Gatter zum Compiler“ als „Durchstich durch viele Abstraktionsebenen“ dienen soll [La15]. Jens Gallenbacher stellt ganz ähnlich einen Unterrichtsgang durch die verschiedenen Abstraktionsebenen vor [vgl. Ga16] und in seinem Buch Abenteuer Informatik [Ga17] beschäftigt er sich mit der Funktionsweise von Computern auf der Ebene von Transistoren und Gattern. Auch die Plattform www.inf-schule.de beschäftigt sich mit *Digitaltechnik* [vgl. Pä16].

Passend zu diesen Unterrichtseinheiten findet man im Internet eine Reihe von Simulatoren für Logikgatter (vgl. etwa [DB05], [Te09], [Bu11]). Digitaltechnik wird also durchaus in bestehendem Informatikunterricht thematisiert.

In Kapitel 1 wurde schon Andreas Schwills Ansatz erwähnt, Bruners Idee der fundamentalen Ideen auf die Informatik zu übertragen [vgl. Kapitel 1]. Bei der Konkretisierung dieser Ideen führt er bzgl. der Hierarchisierung „Ebenenmodelle der Rechnerarchitektur“ [Sc93, S.20] und „boolesche Funktionen“ bei der Orthogonalität [ebd., S.21] auf. Auch die Modularisierung sehen wir bei der Thematik der Rechnertechnik/Rechnerarchitektur gut thematisierbar. Daher dient die Behandlung dieser Thematik als übersichtliches, eingängiges Beispiel dieser fundamentalen Ideen.

³ Vgl. etwa: Hessen [He16 S.40 ff.] und [He05 S. 24f.], Rheinland-Pfalz [Rh17a S.16] und [Rh17b S.24], ansatzweise Thüringen [Th12 S. 18f.] und Berlin [Se06 S.20]

„Der Prozess der Modellierung ist nicht nur Lerninhalt, sondern auch durchgängige Methode des Informatikunterrichts.“, lautet es in den Bildungsstandards Informatik [GI08, S.45]. Auch wenn Modellierung eigentlich durchgängige Methode des Informatikunterrichts sein soll, so ist sie das in der Praxis nicht immer und bei jeder Thematik gegeben, evtl. auch aufgrund der weiteren Anforderungen und Randbedingungen, die in den Bildungsstandards aufgestellt werden: so sei die Implementierung beim informatischen Modellieren „unverzichtbar, um das Ergebnis der Modellbildung erlebbar zu machen.“ [ebd. S.46]. Für die Jahrgangsstufe 5-7 wird aber erklärt: „In diesen Jahrgangsstufen kann der Prozess der Modellierung nicht vollständig durchlaufen werden, da die Fähigkeit zur Abstraktion bei Schülerinnen und Schülern noch nicht hinreichend ausgeprägt ist. Insbesondere wird es kaum möglich sein, in dieser Altersstufe Modelle selbst zu implementieren.“ [ebd.].

Modellierung im Bereich Rechnertechnik, insbesondere auf der Ebene der Logikgatter kann nach Meinung der Autoren des vorliegenden Beitrags diese Problematik umgehen: Digitale Simulatoren oder Stecksysteme für Logikgatter bieten, anders als bei vielen anderen Modellen, die Möglichkeit einer „Quasi-Implementierung“ dieser Modelle, als dass die Funktionsweise des Modells ausprobiert werden kann.

Für die Thematisierung von Hierarchisierung ist der Gang durch mehrere Abstraktionsebenen der Rechnerarchitektur notwendig, bei der Modularisierung hilfreich.

Betrachtet man nur die Ebene der Logikgatter, so kann ein weiterer Kompetenzbereich gefördert werden. Die mathematische Entsprechung für die Logikgatter ist die boolesche Algebra. Boolesche Operatoren sind notwendig zur Konstruktion komplexer Bedingungen in algorithmischen Grundbausteinen. Im Inhaltsbereich Algorithmen der Bildungsstandards wird die Arbeit mit den algorithmischen Grundbausteinen als zentraler Aspekt identifiziert [vgl. GI08, S. 30]. Entscheidungen und Schleifen, außer Zählschleifen, benötigen Bedingungen, welche boolesche Ausdrücke darstellen. Zur Behandlung komplexer Bedingungen werden keine Aussagen gemacht. In vielen Unterrichtszenarien des Anfangsunterrichts mag es auch nicht notwendig sein, den Aufbau dieser Bedingungen intensiv zu thematisieren, weil etwa durch die Blöcke einer visuellen Programmiersprache Vergleiche implizit benutzt werden, oder Rückgabewerte von Methoden direkt einen booleschen Wert darstellen. Auch können durch Schachtelung oder Reihung von if-Strukturen AND und OR-Terme implizit umgesetzt werden. Andererseits kennen selbst kindgerechte Programmiersprachen wie Scratch oder die Lego Mindstorms-Software logische Operatoren. Komplexe logische Bedingungen sind somit auch rein im Kontext der Arbeit mit einer Programmiersprache erlernbar. Auch mittels eines Spieles wie Erkennungsdienst können die logischen Funktionen erarbeitet werden [vgl. Ga17]. Über die Arbeit mit Logikgattern kann darüber hinaus eine Vorentlastung für die Einführung von Bedingungen bei algorithmischen Grundbausteinen oder eine Rückbindung/Wiederholung erfolgen.

3 Quadrologik

Behandelt man Rechnertechnik im Unterricht, sowohl die Ebene einfacher Schaltnetze aus Logikgattern als auch Schaltwerke oder komplexere Strukturen wie ALU, Steuerwerk und Rechenwerk, stellt sich die Frage des *wie*. Wie in Kapitel 2 aufgezeigt, existieren eine Reihe von Simulatoren für Logikgatter.

Werner Hartmann hat zur INFOS2005 erläutert, dass Informatik geprägt sei durch Abstraktion und „[v]ielleicht müssten auch im Informatikunterricht vermehrt die drei klassischen Repräsentationsebenen eingesetzt werden. [...] Abstraktes im wahrsten Sinne des Wortes begreifbar machen [...]“ [Ha05]. Auch die Autoren des vorliegenden Textes sind der Meinung, dass der Unterrichtsgegenstand, wenn möglich, im wörtlichen Sinne begriffen werden sollte. Im Bereich der Rechnertechnik, insb. der Logikgatter, ist es möglich nicht nur die ikonische Repräsentationsebene mittels Simulatoren anzusprechen, sondern eben auch die enaktive Repräsentationsebene durch haptische Systeme. Hierfür existieren verschiedene käuflich zu erwerbende Systeme.

3.1 Vorführsysteme aus der Physik und Steuerungstechnik

Für den Physikunterricht existieren umfangreiche Stecksysteme für Analog- und Digitaltechnik, etwa LOGITRON [GR17], welches aufgrund von Größe und Kosten jedoch eher zur Vorführung durch die Lehrkraft geeignet scheint.

3.2 littleBits

littleBits [sic] ist das Produkt eines New-Yorker Start-Ups. Dabei handelt es sich um modulare Elektronikbauteile, als Bits bezeichnet, welche durch magnetische Konnektoren (bitSnaps) zusammengesteckt werden (vgl. Li17). Zu diesem System gehören auch Logikgatter, die bei littleBits in der Kategorie *WireBits* zusammen mit anderen Verbindungselementen geführt werden. Gegenüber Schaltungssymbolen haben diese Bits jedoch ihre Eingänge nicht an der linken Seite, sondern oben und unten anliegend.

3.3 Quadrologik

Die auf dem Markt vorhandenen Systeme konnten die Autoren des vorliegenden Artikels nicht überzeugen. Daher entwickelt die Arbeitsgruppe ein eigenes Logikstecksystem: *Quadrologik*.

Für dieses System gilt:

- Logikgatter werden, wie allgemein üblich, mit (zwei) Eingängen auf der linken Seite und einem Ausgang auf der rechten Seite eines Bausteines realisiert.⁴
- Verbindungen zwischen Logikgattern werden nicht mit Kabeln gelegt, sondern mit speziellen Verbindungselementen, welche Verbindungen von links nach rechts und oben nach unten implizieren (vgl. unten). Die „Verkabelung“ kann dadurch nicht „kreuz und quer“ erfolgen, sondern muss geplant werden. Darüber hinaus wäre ein derart gelegtes Verbindungsnetz immer auf einer zweilagigen Platine abbildbar.
- Zur Verbindung der Bausteine dienen die Steckverbindungen (bitSnaps) von littleBits. Die Stromversorgung erfolgt über USB(5V). Beides sind Maßnahmen, welche die Kompatibilität zum littleBits-System ermöglichen.
- Die Logikgatter mit zwei Eingängen einerseits und die Verbindungselemente, Taster und das Not-Gatter andererseits werden jeweils durch eine generische Platine realisiert. Erst durch Bestückung bzw. Löt-punkte werden die konkreten Steckelemente festgelegt.

In der bisherigen Entwicklungsstufe sind Schaltnetze möglich.

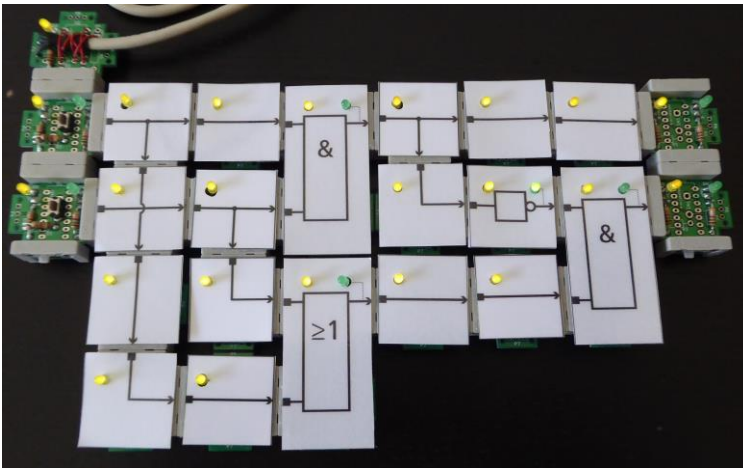


Abb. 1: Halbaddierer mit Quadrologik

4 Erster Einsatz im Unterricht

Der erste Satz Prototypen wurde bisher in unterschiedlichen Unterrichtssituationen getestet.

⁴ Das Not-Gatter besitzt selbstverständlich nur einen Eingang.

Mit mehreren Lerngruppen aus einem fünften Schuljahr (Gymnasium) sowie einer Lerngruppe eines vierten Schuljahres wurde jeweils eine Doppelstunde mit Quadrologik gearbeitet. Aufgrund der bisher begrenzten Anzahl an Quadrologik-Bausteinen wurden jeweils nur 6 Schülerinnen und Schüler aus verschiedenen Klassen „ausgeliehen“. In der Regel wurden dafür von Mathematikkollegen Schülerinnen und Schüler ausgewählt, für die aufgrund ihrer Leistungsstärke die verpasste Mathematik-Einheit entbehrlich war. Für sie stellten diese Stunden ihre ersten Informatikstunden überhaupt dar. Ziel war somit auch, auszuprobieren, was in diesem Anfangsunterricht möglich ist – auch vor dem Hintergrund, dass in dieser Altersstufe noch kein Physikunterricht zu Elektrizität stattgefunden hat.

Als Rahmen für die Stunde wurde ein Anchored Instruction-Szenario gewählt [vgl. GH13]: in der Rolle eines Indiana Jones auf einem fremden Planeten erkundeten die Schülerinnen und Schüler eine alte Ruine, bei der die Funktionsweise von „komischen“ Schalttafeln verstanden werden musste, um die Lampen am Ende der Schaltungen zum Leuchten zu bringen, um dadurch sinnbildlich die Türen einer Ruine zu öffnen.

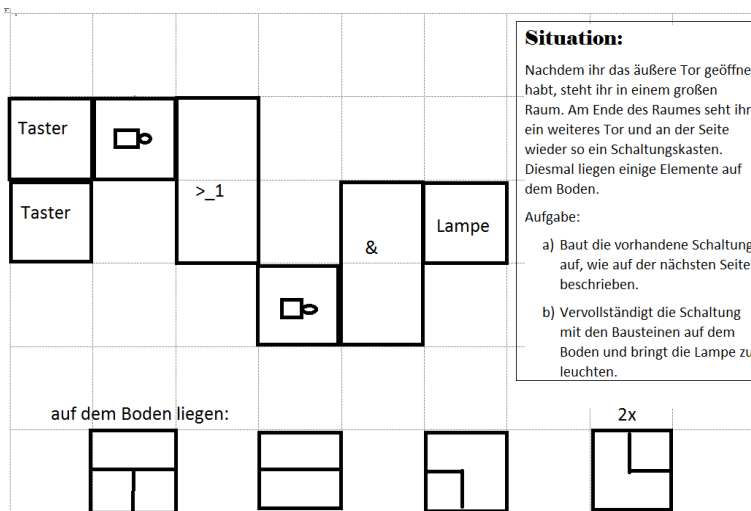


Abb. 2: Aufgabe mit „heruntergefallenen“ Leitungsmodulen

Im Laufe dieser interaktiven Geschichte haben sie sich die Funktionsweise des Stecksystems mit seinen Komponenten und insbesondere der Logikbausteine erschlossen. Bei der ersten Aufgabe gaben wir den Schülerinnen und Schülern eine fertige Schaltung vor. Sie mussten dann die beiden Lampen an den Ausgängen zum Leuchten bekommen. Bei der zweiten mussten sie „heruntergefallene“ Leitungsmodul korrekt in eine teilweise vorgegebene Schaltung einfügen (s. auch Abb. 2). Es folgten Übungen zur Funktionsweise der Logikbausteine wie etwa: „Der Taster lässt sich nicht

drücken“, was die Verwendung eines NOT-Gatters erzwang, um die Lampe am Ausgang leuchten zu lassen (s. auch Abb. 3).

Auf vier Aspekte für die oben benannte Kompetenzentwicklung (vgl. Kapitel 2) wurde geachtet und in den verschiedenen Lerngruppen variiert, auch da in einer Doppelstunde nicht alles behandelbar war.

Mit allen Gruppen gelang die Konstruktion von Wahrheitstabeln (als „An-Aus-Tabelle“). Den leistungsstärksten Gruppen gelang es, die vollständigen Tafeln selbständig aufzustellen. Andere vergaßen zunächst die Variante (Eingang1 aus, Eingang2 aus) mit zu benennen. Spätestens nach der gemeinsamen, exemplarischen Konstruktion einer Tabelle konnten die anderen Tabellen selbstständig hergeleitet werden. Zweimal gab es beim NOT das Problem, dass zunächst auch zwei Eingänge angesetzt wurden. Die Reihenfolge der Zeilen entsprach, erwartungsgemäß, noch nicht der üblichen Konvention.

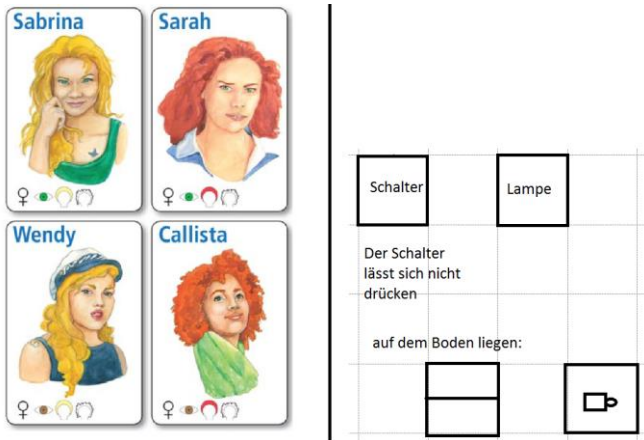


Abb. 3: Spielkarten vom Erkennungsdienst[Ga17] und Aufgabe zum Not

Die Analyse von Schaltungen ist eine Vorstufe zum Erstellen eigener Schaltungsmodelle (vgl. [GI08, S.46]). Durch Nachbau der Schaltungen und Ausprobieren aller Kombinationen konnten die Schülerinnen und Schüler selbstständig vollständige Wahrheitstabellen für drei Eingänge aufstellen. Der theoretische Nachvollzug anhand des Schaltungsplans wurde mit einer Gruppe versucht. Hier zeigte sich, dass das Verständnis des Schaltungsaufbaus noch nicht vollständig durchdrungen war. So konnten die Schülerinnen und Schüler zwar benennen, wie in einer mehrstufigen Logik die Signalleitungen verlaufen, sie erkannten aber nicht selbstständig, dass die Eingänge eines späteren Logikgatters nicht direkt von den Eingängen/Tastern abhängen, sondern dass sie an den Ausgängen der ersten Logikelemente Zwischenergebnisse ermitteln müssen.

Um den Zusammenhang mit logischen Aussagen zu bekommen, wurde mit Spielkarten

des Spiels Erkennungsdienst [vgl. Ga17] gearbeitet (s. auch Abb. 3). Zunächst wurden logische Aussagen über die Merkmale der Personen vorgegeben (etwa: rothaarig UND grüne Augen), die Schülerinnen und Schüler mussten dann die entsprechende(n) Karte(n) benennen. Ein Teil hatte anfangs Probleme, mehr als eine Karte zu benennen, wenn eine Teilmenge von mehr als einer Karte gefordert war. Dieses Problem war allerdings schnell behoben. Der Umgang mit dem UND-Operator war erwartungsgemäß unproblematisch. Aber auch Aussagen mit der NICHT-Operator und das inklusive ODER funktionierten gut. Im Rahmen der Wahrheitstabellen wurde thematisiert, dass ein Unterschied zwischen mathematischer Logik und Alltagssprache besteht.

Im zweiten Schritt durften die Schülerinnen und Schüler logische Aussagen für Teilmengen der Karten formulieren (z.B.: „Erstelle eine Aussage, welche nur auf Kevin und Felix zutrifft“). Dies klappte erstaunlich gut. Einzelne Schülerinnen und Schüler kamen auf Ausdrücke, die mathematisch nur mit Klammerung formulierbar wären. Dies wurde im Einzelgespräch erfolgreich thematisiert. Die Schülerinnen und Schüler waren danach in der Lage, gegebenen logischen Aussagen die entsprechenden korrekten Teilmengen an Personenkarten zuzuordnen.

Das eigentlich mit Quadrologik verfolgte Ziel, die Konstruktion einer eigenen Schaltung, wurde zuletzt erprobt. Bei einer Lerngruppe wurde die Konstruktion einer Schaltung gefordert, für welche nur eine Wertetabelle vorgegeben war. Einer Gruppe gelang es NAND und NOR Schaltungen sowie eine weitere Funktion (ein Und-Gatter mit einem negierten Eingang) zu konstruieren, scheiterten dann jedoch am XOR, allerdings auch aus Zeitgründen. Bei drei weiteren Lerngruppen wurde im Rahmen des Rollenspiels die Aufgabe gestellt, dass ein bestimmter Roboter in Menschengestalt durch eine Einlasskontrolle passieren darf und die anderen Roboter nicht. Dazu wurde eine aus acht Erkennungsdienstkarten gezogen. Sämtlichen Gruppen gelang es, diese Aufgabe zu meistern. In der Menge der acht Karten waren die Merkmale so verteilt, dass für die Selektion einer Karte alle drei Merkmale entweder regulär oder negiert abgefragt und mit UND verknüpft werden mussten. Aufgrund dessen der lediglich zweiwertigen Logikbausteine bei Quadrologik mussten also alle eine mehrstufige Logik mit geschachtelter Aussage realisieren.

Auch auf der subjektiven Ebene waren die Unterrichtsversuche erfolgreich. So erklärten alle SuS, dass der Unterricht ihnen Spaß gemacht hätte, fanden das Quadrologik-System sehr toll und wollten mehr Unterricht dazu.

Im Wahlunterricht Informatik im Jahrgang 9 wurde Digitaltechnik „klassisch“ unterrichtet. Der Teil mit Logikgattern wurde diesmal jedoch mit Quadrologik ausgeführt. Die Schüler⁵ besaßen vertieftes Vorwissen, und alle nahmen auch schon an der Juniorliga des BWInf teil. In 2,5 Doppelstunden wurde die Thematik von der Einführung von Logikgattern, über die Analyse komplexer Schaltung bis zum Entwurf eigener Schaltungen behandelt. Für die Modellierung verschiedener Schaltungen bekamen die Schüler eine Sammlung diverser Textaufgaben zu Logikgattern, von

⁵ Der Kurs besteht leider nur aus männlichen Individuen.

Tresorsteuerungen über Gurtkontrollen im Auto bis hin zu Multiplexern und Halb-/Volladdierern. Die Bearbeitung dieser Problembeschreibungen gelang gut bis sehr gut, entsprechend den allgemeinen Leistungsunterschieden im Kurs. Beim Entwurf des Volladdierers war die Analyse der Problemstellung schnell erfolgt, eigene Wahrheitstafeln aufgestellt und der Baustein mit Quadrologik gesteckt (s. auch Abb. 1)

5 Ausblick

Rechnertechnik sollte im Informatikunterricht behandelt werden, um dabei Modellbildungskompetenz und Verständnis von Modularisierung zu fördern. Das gelingt mit dem Logikstecksystem Quadrologik auf motivierende Art und Weise. Erste Tests mit Schülergruppen sind sehr vielversprechend. Es zeigte sich, dass der erhoffte Kompetenzzuwachs erreicht wird.

Wir möchten die Weiterentwicklung von Quadrologik betreiben. Dazu gehört die Realisierung von Schaltnetzen und die Unterstützung der Modularisierung durch ein eigenes Bauteil, sodass konstruierte Schaltungen in diese Bauteile einprogrammiert werden können. Wir erhoffen uns Impulse für die konkrete Ausgestaltung der Weiterentwicklung aus der Diskussion auf der INFOS und danach.

Sofern diese Entwicklungsprozesse abgeschlossen sind, können eine größere Anzahl an Bauteilen produziert werden, sodass Tests in Klassenstärke und über unterschiedliche Kompetenzstufen sowie eine wissenschaftliche Evaluation möglich sind.

Literaturverzeichnis

- [Bu11] Burch, C.: Logisim v.2.7.1., <http://www.cburch.com/logisim/>, Stand 05.02.2017.
- [DB05] Damm, M; Brauner, O.: LogiFlash v3.03, 2005
<http://tiweb.hsu-hh.de/LogiFlash/index.html>, Stand 05.02.2017.
- [Ga16] Gallenbacher, J.: Rechnerorganisation – Vom Relais zum Prozessor. Vortrag beim Schweizer Tag für den Informatikunterricht. Zugriff über: <http://www.abz.inf.ethz.ch/schweizer-tag-fur-den-informatikunterricht/stiu-2016-7-schweizer-tag-fuer-den-informatikunterricht/stiu-2016-workshops/>, Stand 09.02.2017
- [Ga17] Gallenbacher, J: Abenteuer Informatik: IT zum Anfassen für alle von 9 bis 99 – vom Navi bis Social Media, 4. Auflage, 2017.
- [GH13] Gallenbacher, J., Heun, D.: Ein moderner Ansatz für Anchored Instruction im Informatikunterricht. In (Breier, N., Stechert, P., Wilke, T. Hrsg.): Proc. Informatik erweitert Horizonte, INFOS2013, 15. GI-Fachtagung Informatik und Schule, 2013.
- [GI08] Gesellschaft für Informatik e.V.: Grundsätze und Standards für die Informatik in der Schule: Bildungsstandards Informatik für die Sekundarstufe I, 2008.
- [GR17] GRS Lehrgeräte KG: Logitron – Digitalbausteine mit starken Eigenschaften,

http://www.grs-physik-shop.de/index.php?option=com_content&task=view&id=37&Itemid=60, Stand 05.02.2017.

- [Ha05] Hartmann, W.: Informatik – EIN/AUS – Bildung. In (Friedrich, Srefflen Hrsg.): Proc. Unterrichtskonzepte für informatische Bildung, INFOS2005, 11. GI-Fachtagung Informatik und Schule, 2005.
- [He05] Hessisches Kultusministerium: Berufliche Schulen des Landes Hessen. Lehrplan Berufliches Gymnasium. Fachrichtung Technik. Schwerpunkt Datenverarbeitungstechnik, 2005.
https://kultusministerium.hessen.de/sites/default/files/media/hkm/lp_bg_fr_technik_sp_datenverarbeitungstechnik_05.pdf, Stand 09.02.2017
- [He16] Hessisches Kultusministerium: Kerncurricula Gymnasiale Oberstufe – Informatik, 2016. <https://kultusministerium.hessen.de/schule/kerncurricula/gymnasiale-oberstufe/informatik>, Stand: 05.02.2017.
- [La15] Lautebach, U.: Vom Gatter zum Compiler: Im Unterricht durch sieben Abstraktionsebenen. In (Gallenbacher, J. Hrsg.): Proc. Informatik allgemeinbildend begreifen, INFOS2015, 16. GI-Fachtagung Informatik und Schule, S.239-246.
- [Li17] LittleBits Electronics Inc.: What is littleBits? <http://littlebits.cc/how-it-works>, Stand 05.02.2017.
- [Pä16] Pädagogisches Landesinstitut Rheinland-Pfalz: Digitaltechnik, 2016, <http://www.inf-schule.de/rechner/digitaltechnik>, Stand: 05.02.2017.
- [Rh17a] Rheinland-Pfalz Ministerium für Bildung, Wissenschaft, Jugend und Kultur. Lehrplan Informatik Wahlfach und Wahlpflichtfach an Gymnasien und Integrierten Gesamtschulen (Sekundarstufe I), ohne Jahresangabe. Zugriff über: <http://lehrplaene.bildung-rp.de/gehezu/startseite.html?keyword=informatik>, Stand 09.02.2017.
- [Rh17b] Rheinland-Pfalz Ministerium für Bildung, Wissenschaft, Jugend und Kultur. Lehrplan Informatik Grund- und Leistungsfach, ohne Jahresangabe. Zugriff über: <http://lehrplaene.bildung-rp.de/gehezu/startseite.html?keyword=informatik>, Stand 09.02.2017.
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik, 1993. <http://ddi.cs.uni-potsdam.de/Forschung/Schriften/ZDM.pdf>, Stand 09.02.2017
- [Se06] Senatsverwaltung für Bildung, Jugend und Sport Berlin: Rahmenlehrplan für die gymnasiale Oberstufe – Informatik, 2006.
https://www.berlin.de/imperia/md/content/sen-bildung/unterricht/lehrplaene/sek2_informatik.pdf?start&ts=1450262874&file=sek2_informatik.pdf Stand: 10.02.2017.
- [Te09] Tetzl, A.: LogicSim, 2009, http://www.tetzl.de/java_logic_simulator.html, Stand 05.02.2017.
- [Th12] Thüringer Ministerium für Bildung, Wissenschaft und Kultur: Lehrplan für den Erwerb der allgemeinen Hochschulreife – Informatik, 2012. Zugriff über: <https://www.schulportal-thueringen.de/media/detail?tspi=3657> Stand 09.02.2017

Integration des Erwerbs von Basiskonzepten der Informatik in den mathematisch-naturwissenschaftlichen Unterricht der Sekundarstufe I

Marlene Lindner¹, Sandra Schulz¹ und Niels Pinkwart¹

Abstract: Als Beitrag zu einer flächendeckenden informatischen Bildung kann sich eine Verknüpfung der Informatik mit anderen Schulfächern gewinnbringend auswirken. Mit Unterrichtseinheiten in *Physical Computing* ist es möglich den Erwerb von Basiskonzepten und -kompetenzen der Informatik in den mathematisch-naturwissenschaftlichen Unterricht zu integrieren. In diesem Beitrag werden unter Berücksichtigung der Bildungsstandards der Kultusministerkonferenz und der Gesellschaft für Informatik Verknüpfungsmöglichkeiten für die Sekundarstufe I aufgezeigt, die sich aus den Bildungsstandards ergeben. Anschließend werden zwei entwickelte Unterrichtseinheiten mit LEGO Mindstorms detailliert vorgestellt.

Keywords: MINT-Unterricht, Bildungsstandards, *Physical Computing*, LEGO Mindstorms.

1 Einleitung

1.1 Fächerübergreifender MINT-Unterricht

Im Allgemeinen kann ein fächerübergreifender Unterricht genutzt werden, um verschiedene Zugänge zu einer Thematik zu ermöglichen. Damit können bisher bestehendes Wissen gefestigt sowie zusammengehörige Inhalte miteinander verknüpft werden.

Kompetenzen in Mathematik, Informatik, Naturwissenschaft und Technik (MINT) sind in den letzten Jahren zunehmend als Schlüsselqualifikationen für den Einstieg in vielfältige Berufsmöglichkeiten identifiziert worden und eine grundlegende *Scientific Literacy* ist mittlerweile ein erklärtes Bildungsziel für SchülerInnen aller Schulformen [OE99, NG13]. Obwohl MINT-Fächer Gemeinsamkeiten in Bezug auf Problemlösung, Erkenntnisgewinnung und lebensweltliche Anwendung haben, werden sie in den meisten Curricula-Dokumenten der Welt getrennt aufgeführt [BB14]. Betrachtet man jedoch die Bildungsstandards der Kultusministerkonferenz (KMK) für die Mathematik und die Naturwissenschaften bzw. die Standards der Gesellschaft für Informatik (GI) und des Vereins Deutscher Ingenieure (VDI) für die Informatik und die Technik, ergeben sich viele Verknüpfungsmöglichkeiten sowohl in Bezug auf den Erwerb von Basiskonzepten als auch Basiskompetenzen. Im Folgenden wird nun speziell auf inhaltliche und konzeptuelle Überschneidung der Informatik mit den mathematisch-naturwissenschaftlichen Fächern eingegangen und Unterrichtseinheiten in *Physical Computing* präsentiert, die sowohl einen Lernzuwachs bzgl.

¹ Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, marlene.lindner@hu-berlin.de; saschulz@informatik.hu-berlin.de; niels.pinkwart@hu-berlin.de

Informatik als auch einem der weiteren Fächer ermöglichen sollen. *Physical Computing* beschäftigt sich mit interaktiven, physischen Objekten, in denen mithilfe von Sensoren und Aktuatoren ein Bezug zwischen virtueller und realer Welt hergestellt wird [OI04]. Zusätzlich bietet es die Möglichkeit, Informatik in den MNT-Unterricht zu integrieren [SP15].

Für Mathematik, Physik und Technik gibt es viele Unterrichtsbeispiele, die eine Verknüpfung zur Informatik herstellen und über die bloße Anwendung von Software hinausgehen. Es wurden jedoch keine Unterrichtsbeispiele gefunden, die explizit die deutschen Bildungsstandards berücksichtigen. LEGO Education bietet ausgearbeitete Unterrichtseinheiten für LEGO Mindstorms EV3-Roboter zum Verkauf an, die Inhalte aus Mathematik, Physik und Technik aufgreifen [Th16]. Vernier Software & Technology [Ve16] bietet eine Vielzahl an Sensoren an, um den Roboter für den mathematisch-naturwissenschaftlichen Unterricht durch Inhalte wie z. B. Messungen des pH-Werts oder eines magnetischen Felds zu erweitern. Zusammenfassend lässt sich feststellen, dass frei verfügbare, deutschsprachige Unterrichtseinheiten, die als Ziel haben Informatik fächerbergreifend mit den MNT-Disziplinen zu unterrichten und die Bildungsstandards berücksichtigen, bisher kaum vorliegen. Daher werden im Folgenden mögliche Verknüpfungspunkte diskutiert und entsprechende Unterrichtseinheiten exemplarisch vorgestellt.

1.2 Bildungsstandards und Rahmenlehrpläne der MINT-Fächer

Seit 2003 liegen von der KMK „Bildungsstandards im Fach Mathematik für den Mittleren Schulabschluss“ vor [KMc] und seit 2004 gibt es auch Bildungsstandards für Biologie, Chemie und Physik [KMa, KMb, KMd]. Sie finden starke Berücksichtigung in den jeweiligen Rahmenlehrplänen der Bundesländer. Für den Informatikunterricht an deutschen Schulen gibt es keine von der KMK verfassten Bildungsstandards. Bisher wurde ein erster Ansatz im Strategiepapier „Bildung in der digitalen Welt“ formuliert, das jedoch nicht konkret auf den Informatikunterricht eingeht [KM16]. Die GI hat 2008 unter „Grundsätze und Standards für die Informatik in der Schule“ [Ge08] sogenannte Mindeststandards für die Sekundarstufe I veröffentlicht, die als methodische und inhaltliche Empfehlung für den Informatikunterricht z. B. in dem aktuellen Rahmenlehrplan von Berlin berücksichtigt sind [Se06]. Da die KMK auch keine Bildungsstandards für das Fach Technik veröffentlicht hat, hat der Verein Deutscher Ingenieure (VDI) 2007 „Bildungsstandards Technik für den Mittleren Schulabschluss“ konzipiert [VD07]. Bisher konnte sich in den Bildungsstandards und Rahmenlehrplänen der MINT-Fächer noch keine einheitliche Terminologie für grundlegende Konzepte und Kompetenzen durchsetzen. Im Folgenden wird der Begriff „Basiskonzepte“ für Inhaltsbereiche, inhaltsbezogene Kompetenzen, Kompetenzbereich Fachwissen und Leitideen verwendet, wohingegen mit „Basiskompetenzen“ Prozessbereiche, allgemeine Kompetenzen und die Kompetenzbereiche Erkenntnisgewinnung, Kommunikation und Bewertung beschrieben werden. Im Rahmen dieses Beitrags kann jedoch nicht näher auf die Verknüpfungspunkte in den Bildungsstandards bzgl. der Basiskompetenzen eingegangen werden.

2 Mögliche Verknüpfungspunkte in den Bildungsstandards bzgl. der Basiskonzepte

Als Basiskonzepte der **Informatik** hat die GI folgende fünf Bereiche identifiziert: „Information und Daten“ (C1), „Algorithmen“ (C2), „Sprachen und Automaten“ (C3), „Informatiksysteme“ (C4) und „Informatik, Mensch und Gesellschaft“ (C5). Im Folgenden werden exemplarisch berschneidungspunkte zu MN-Fchern aufgezeigt, die in Tab. 1 zusammengefasst sind. Fr den **Mathematik**-Unterricht lassen sich die Konstruktion und Berechnung von geometrischen Strukturen und das kartesische Koordinatensystem in „Raum und Form“ (L3) mit C1 vernetzen. Dies sind Grundlagen für das Erfassen von Daten und Informationen in Grafiken und Graphen. Dafr kann ein Grafikprogramm Anwendung finden. In der **Biologie** ist ein wesentlicher Aspekt des Basiskonzepts „System“ (F1) die Steuerung, Regelung und Informationsverarbeitung in lebenden Systemen. Es lsst sich mit (C3) z. B. durch das EVA-Prinzip und Zustnde der Automatentheorie verknpfen. „Entwicklung“ (F3) beinhaltet die Auseinandersetzung mit artspezifischer Individualentwicklung und evoluti-onärer Entwicklung, wobei Mutation, Selektion und Variabilität behandelt werden. Das Beschreiben der Eigenschaften bedeutsamer Stoffe in der **Chemie** ist ein Kernthema von „Stoff-Teilchen-Beziehungen“ (F1), wohingegen in „Struktur-Eigenschafts-Beziehungen“ (F2) das Ableiten von Stoffeigenschaften aus dessen Struktur betrachtet wird. In „Chemische Reaktion“ (F3) verlangt das Erstellen von ausgeglichenen Reaktionsgleichungen (F3.4) quantitatives Verständnis, was auch für die daraus abgeleiteten Handlungsvorschriften für Experimente relevant ist. „Energetische Betrachtung bei Stoffumwandlung“ (F4) betrachtet Energieformen und Energieaustausch mit der Umgebung. Energieumwandlung und die Abhängigkeit von Reaktionsbedingungen und Katalysatoren (F4.3) ist auch relevant für einige Komponenten in Informatiksystemen (C4), wie z. B. Sensoren oder Elemente zur Kühlung von Rechneranlagen. Das Materiekonzept in **Physik** lässt sich lediglich mit C4 in Bezug auf Hardwarekomponenten von Informatiksystemen und deren Kenngrößen verbinden. Das Wechselwirkungskonzept lässt sich hingegen mit fast allen Basis-kompetenzen der Informatik verknüpfen. Die newtonsche Mechanik, Optik, sowie elektrische und magnetische Felder sind für die Eingabe-, Verarbeitungs- und Ausgabegeräte von Informatiksystemen (C4) von Bedeutung, beeinflussen aber auch Wechselwirkungen zwischen Informatik, Mensch und Gesellschaft (C5).

3 Exemplarische Unterrichtseinheiten im Physical Computing

Aufgrund der herausgearbeiteten berschneidungen erscheint Physical Computing als geeigneter Lerngegenstand, um eine Verknüpfung sinnvoll umzusetzen. Folgende Beispiele für Unterrichtseinheiten im *Physical Computing* sollen es ermöglichen, den Erwerb von Basiskonzepten und -kompetenzen der Informatik in den naturwissenschaftlichen Unterricht zu integrieren. Dabei geht es nicht darum, besonders vielschichtige, komplexe Aufgaben zu stellen, sondern vielmehr Beispiele aufzuzeigen, die sowohl in den normalen Fachunterricht als auch in den Projektunterricht eingebunden werden können und in Bezug auf Zeit und Material keine großen Herausforderungen darstellen. Auf Sensoren, die nicht in der Standardverkaufseinheit der LEGO Mindstorms EV3-Roboter enthalten

Informatik	Mathematik	Biologie	Chemie	Physik
Information und Daten (C1)	Zahl (L1), Messen (L2), Raum und Form (L3), Funktionaler Zusammenhang (L4), Daten und Zufall (L5)	System (F1), Struktur und Funktion (F2), Entwicklung (F3)	Energie (F4), Stoff-Teilchen-Beziehungen (F1), Struktur-Eigenschafts-Beziehungen (F2)	Wechselwirkung, Energie
Algorithmen (C2)	Zahl (L1), Raum und Form (L3), Funktionaler Zusammenhang (L4), Daten und Zufall (L5)	System (F1), Struktur und Funktion (F2)	Stoff-Teilchen-Beziehungen (F1), Struktur-Eigenschafts-Beziehungen (F2), Chemische Reaktion (F3)	Wechselwirkung, System
Sprachen und Automaten (C3)	Zahl (L1), Funktionaler Zusammenhang (L4)	System (F1), Struktur und Funktion (F2)	Chemische Reaktion (F3)	System
Informatiksysteme (C4)	Funktionaler Zusammenhang (L4)	System (F1), Struktur und Funktion (F2), Entwicklung (F3)	Energie (F4), Stoff-Teilchen-Beziehungen (F1), Struktur-Eigenschafts-Beziehungen (F2)	Materie, Wechselwirkung, System, Energie
Informatik, Mensch und Gesellschaft (C5)	Raum und Form (L3), Funktionaler Zusammenhang (L4), Daten und Zufall (L5)	System (F1), Struktur und Funktion (F2), Entwicklung (F3)	Energie (F4), Stoff-Teilchen-Beziehungen (F1), Struktur-Eigenschafts-Beziehungen (F2), Chemische Reaktion (F3)	Wechselwirkung, Energie

Tab. 1: Verknüpfung der Basiskonzepte (Inhaltsbereiche) der Bildungsstandards Informatik für die Sekundarstufe I [Ge08] mit Basiskonzepten (inhaltsbezogenen Kompetenzen, Kompetenzbereich Fachwissen (F), Leitideen (L)) der Bildungsstandards im Fach Mathematik, Biologie, Chemie und Physik für den Mittleren Schulabschluss [KMc, KMa, KMb, KMd].

sind, wird bewusst verzichtet. Auch wird auf die Umsetzbarkeit jeder Unterrichtseinheit in einer Doppelstunde (90 min) geachtet, wobei jeweils insgesamt 10 min für Unterrichtsbeginn und -abschluss veranschlagt werden, da die Organisation von Gruppen und Material berücksichtigt werden muss. Es wird davon ausgegangen, dass genügend Roboter vorhanden sind um jeweils zu zweit mit einem Roboter zu arbeiten. Bei den meisten Aufgaben bietet es sich an, mit bereits zusammengebauten Robotern zu arbeiten. Für einen selbstständigen Zusammenbau müssten zusätzlich ca. 30 min eingeplant werden. Weitere Unterrichtsbeispiele der MINT-Fächer sowie Unterrichtsmaterialien werden unter https://cses.informatik.hu-berlin.de/de/for_schools/ zur Verfügung gestellt.

3.1 Unterrichtseinheit in der Biologie: Informationsverarbeitung

Klassenstufe:	9 (abhängig von der Schulform)
Vorwissen in Informatik:	Kenntnisse von algorithmischen Grundstrukturen
Informatische Inhalte:	Roboter als Informatiksysteme, Implementieren von Algorithmen, Verarbeitung elektrischer Größen (<i>Physical Computing</i>)
Robotertyp:	Standardfahrzeug der LEGO Mindstorms EV3-Roboter mit zwei Lichtsensoren
Zusätzliches Material:	Taschenlampen

Tab. 2: Aufgabenprofil der Unterrichtseinheit „Informationsverarbeitung“

Für alle Organismen ist die Verarbeitung von Informationen aus ihrer Umwelt von zentraler Bedeutung für Nahrungs- und Partnersuche. Sogenannte Reiz-Reaktions-Zusammenhänge sind mitunter sehr komplex, abhängig von vielen Faktoren und somit schwer zu verstehen (vgl. EVA-Prinzip der Informatik). Braitenberg-Vehikel hingegen haben eine sehr einfache Struktur (Abb. 1), verdeutlichen aber die Vielfalt, die schon mit einfachen neuronalen Strukturen erzeugt werden kann. Zwei Sensoren werden auf unterschiedliche Weise (inhibitorisch oder exzitatorisch) mit zwei Motoren gekoppelt und erzeugen somit ein „Verhalten“, das unterschiedlichen Emotionen bzw. Verhaltensstrategien zugeordnet werden kann. Mit LEGO Mindstorms EV3-Robotern können Braitenberg-Vehikel nachgebaut und beobachtet werden. Der Fokus dieser Unterrichtseinheit bzgl. Biologie liegt somit auf der Erklärung der Variabilität von Lebewesen (Basiskonzept „Entwicklung“ (F3)) durch den Einsatz eines Modells (Basiskompetenz E10). Innerhalb der Informatik wird insbesondere das Verstehen und Anwenden von Informatiksystemen adressiert. Schleifen und Schalter werden als bekannt voraus gesetzt. Bei Aufgabe 1 ist abhängig vom Umgebungslicht ein geeigneter Schwellwert für den Lichtsensor zu finden. Als Programmierblock für den Motor sollte *Hebelsteuerung* mit *An für n Sekunden* und *Am Ende Bremsen: Falsch* gewählt werden. Rauschen ist ein wichtiger Aspekt bei der Informationserfassung und -verarbeitung, sowohl in der Technik als auch in lebenden Organismen. In diesem Zusammenhang sollte kurz auf das Nervensystem des Menschen eingegangen werden und erläutert werden, dass Synapsen auch zufällig feuern, wodurch sich ein Rauschen bei der Reizübertragung ergibt. Des Weiteren kann bei der Verarbeitung elektrischer Größen auf Messunsicherheiten eingegangen und Algorithmen zur Fehlerminimierung entwickelt werden. Im Rahmen von Aufgabe 2 kann Rauschen mit dem Programmierblock

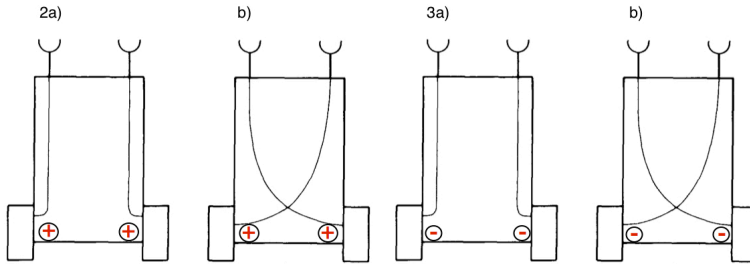


Abb. 1: Braitenberg-Vehikel als Modellkonstrukte für die Informationsverarbeitung in einfachen Organismen: Zwei Sensoren werden auf unterschiedliche Weise (inhibitorisch oder exzitatorisch) mit zwei Motoren gekoppelt und erzeugen somit ein „Verhalten“, das unterschiedlichen Emotionen zugeordnet werden kann. 2a) Angst b) Aggression; 3a) Liebe b) Entdeckungslust [Br84].

Zufall simuliert werden. Bei der Chemotaxis z. B. peritrich begeißelter Bakterien wechseln sich Vorwärtsbewegung entlang eines Stoffkonzentrationsgradienten und zufällige Tauselbewegungen ab, die eine Richtungsänderung ermöglichen, wenn die Stoffkonzentration sinkt. Die Geschwindigkeit bei der Phototaxis soll bei Aufgabe 3 proportional zu der Lichtintensität steigen – dies ist ein Braitenberg-Vehikel vom Typ I [Br84]. Für eine ef-

Aufgabe	Zeit
1. Lasst euren Roboter geradeaus auf euch zufahren, wenn ihr ihn mit einer Taschenlampe anstrahlt. Sobald die Taschenlampe aus ist, soll er anhalten und warten. Verwendet zunächst nur einen der Lichtsensoren.	10 min
2. Lasst euren Roboter sich zufällig leicht nach rechts und links bewegen, wenn er wie unter 1. auf euch zu fährt.	10 min
3. Lasst euren Roboter nun schneller auf euch zufahren, wenn ihr ihn direkt anstrahlt und langsamer werden, wenn die Lichtquelle weiter entfernt ist. Vergleicht das Verhalten eures Roboters mit der Chemotaxis von Bakterien. Was muss für eine effektive Phototaxis des Roboters verändert werden?	15 min
4. Baut euer Programm nun so um, dass beide Lichtsensoren gleichzeitig messen und jeweils einen der Motoren ansteuern. Testet, ob sich eurer Roboter nun Lichtquellen zuwendet, die nicht direkt vor ihm sind.	15 min
5. Schreibt nun ein Programm für ein weiteres Braitenberg-Vehikel und testet seine Reaktion auf eine Lichtquelle. Von welchen Tieren würdet ihr eine vergleichbare Reaktion erwarten?	15 min
6. Dunkelt den Raum nun ab und schafft eine große freie Fläche, auf der ihr eure Braitenberg-Vehikel aussetzt. Erzeugt einen Lichtkegel etwa in der Mitte der Fläche und beobachtet das Verhalten aller Roboter. Ordnet ihr Verhalten jeweils Typen von Braitenberg-Vehikel zu und diskutiert Verbesserungsmöglichkeiten bei der Umsetzung. Ändert ggf. das Programm eures Roboters und setzt ihn erneut auf die Freifläche.	15 min

Tab. 3: Unterrichtseinheit „Informationsverarbeitung“: Arbeitsaufgaben und Durchführungsdauer

effektive Phototaxis fehlt noch die Implementation einer Bewegungsrichtungsänderung. Für Aufgabe 4 gibt es mehrere Lösungsmöglichkeiten. Zwei parallele Schleifen mit jeweils einem einzelnen Motor lassen sich hierbei gut umsetzen. Dies ist nun das „aggressive“ Braitenberg-Vehikel (Typ 2b). Die Braitenberg-Vehikel sollten mit Abb. 1 kurz vor-

gestellt werden und bei Aufgabe 5 sollte jeder Vehikel-Typ von mindestens einer Gruppe bearbeitet werden, wobei der Typ 2a) sehr einfach aus 2b) ableitbar ist. Durch die inhibitorische oder exzitatorische Verknüpfung der zwei Sensoren mit den beiden Motoren können vier unterschiedliche Verhaltensmuster erzeugt werden: einige Roboter fliehen vor dem Licht, andere greifen den Lichtkegel an, einige bleiben darin stehen und wieder andere betrachten den Lichtkegel kurz und ziehen dann weiter. Diese Verhaltensmuster werden mitunter Emotionen zugeordnet. Hier ist jedoch für einen sprachbewussten Biologieunterricht wichtig, den Unterschied zwischen Verhalten und Emotionen zu verdeutlichen und im Ausdruck auf das Vermeiden von Begriffen zur Vermenschlichung der Roboter zu achten. Tiere, die mitunter vergleichbares Verhalten zeigen, sind z. B. Kellerassel (2a), Motte (2b), Eidechse (3a) und Katze (3b). Bei Aufgabe 6 werden sich Zusammenstöße nicht ganz vermeiden lassen und die Typen 2a) und 3b) müssen eingefangen werden, bevor sie in dunklen Ecken des Raumes verschwinden bzw. gegen die Wand fahren.

3.2 Unterrichtseinheit in der Chemie: Photometrie

Klassenstufe:	9 (abhängig von der Schulform)
Vorwissen in Informatik:	Kenntnisse von algorithmischen Grundstrukturen, Variablen und Arrays
Informatische Inhalte:	Roboter als Informatiksysteme, Arrays in der Messwerterfassung, Implementieren von Algorithmen, <i>Physical Computing</i>
Robotortyp:	LEGO Mindstorms EV3-Stein mit zwei Lichtsensoren
Zusätzliches Material:	Küvetten und Küvettenhalterung, Lebensmittelfarbe (z. B. E132), Wasser, Pipetten, Bechergläser

Tab. 4: Aufgabenprofil der Unterrichtseinheit „Photometrie“

Die Absorption von Licht durch farbige Lösungen ist ein Alltagsphänomen, das mit Hilfe von photometrischen Untersuchungen quantifiziert werden kann. Der Transmissionsgrad τ ergibt sich aus dem Quotienten der Intensität des Lichtstrahls vor (I_0) und nach (I_1) dem Durchtritt durch die Lösung. Die Extinktion E ergibt sich aus dem Logarithmus des Kehrwerts von τ und ist proportional zu der Konzentration c des lichtabsorbierenden Stoffes und der Weglänge d des Lichtstrahls in der Lösung. Der Proportionalitätsfaktor ε wird als stoffspezifischer, molarer Extinktionskoeffizient bezeichnet. So ergibt sich das Lambert-Beersche Gesetz mit:

$$E = \log \left(\frac{I_0}{I_1} \right) = \varepsilon \cdot c \cdot d \quad (1)$$

Der LEGO Mindstorms EV3-Stein kann verwendet werden um photometrische Messungen durchzuführen. Hierbei wird die duale Eigenschaft des Lichtsensors ausgenutzt, der sowohl Lichtintensitäten detektieren, als auch rotes Licht einer Wellenlänge von $632 \text{ nm} \pm 3 \text{ nm}$ erzeugen kann. Diese Unterrichtseinheit orientiert sich an dem Artikel „Using LEGO MINDSTORMS NXT™ Robotics Kits as a Spectrophotometric Instrument“ [KWB10], in dem Messungen mit dem Lichtsensor der LEGO Mindstorms mit denen eines *Hitachi* U-2000 Spektrophotometer verglichen wurden. LEGO Mindstorms können nicht verwendet werden um ε korrekt zu bestimmen, jedoch können Stoffkonzentrationen relativ genau ermittelt und somit das Lambert-Beersche Gesetz im Experiment verdeutlicht werden.

Der Fokus dieser Unterrichtseinheit bzgl. Chemie liegt somit auf der Durchführung einfacher qualitativer und quantitativer experimenteller Untersuchungen (Basiskompetenz E3) und der Modellnutzung zur Deutung der Lichtabsorption eines Farbstoffs (Basiskonzept „Struktur-Eigenschafts-Beziehungen“ (F2)). Indigocarmin ist als Lebensmittelfarbstoff

Aufgabe	Zeit
1. Nehmt euch 10 ml der Indigocarmin-Stammlösung, 10 Küvetten, destilliertes Wasser, eine Pipette und ein leeres Becherglas. Setzt eine Verdünnungsreihe an und füllt jeweils 1 ml in eine Küvette. Notiert euch die Mischungsverhältnisse! Eine der Küvetten soll als Leerprobe nur mit destilliertem Wasser gefüllt werden.	10 min
2. Baut aus LEGO-Steinen eine Küvettenhalterung mit zwei Lichtsensoren und schließt die Sensoren an euren LEGO Mindstorms EV3-Stein an.	5 min
3. Schreibt ein Programm, bei dem die Intensität des Umgebungslichts von einem der Lichtsensoren mehrfach gemessen, in einem Array gespeichert und auf dem Stein-Display angezeigt wird. Verwendet die Steintasten um euch die Werte nacheinander anzeigen zu lassen.	15 min
4. Verändert euer Programm nun so, dass der zweite Lichtsensor permanent rotes Licht ausstrahlt. Notiert euch die gemessenen Intensitätswerte ohne Probe und messt nun eine Probe mit mittlerer Verdünnung.	10 min
5. Vergleicht eure Messwerte mit denen der anderen Gruppen und diskutiert in Anbetracht der Messwertschwankungen über eine geeignete Zahl von Wiederholungsmessungen.	5 min
6. Erweitert euer Programm nun so, dass ihr alle Proben nacheinander mehrfach vermessen könnt, ohne das Programm neu starten zu müssen. Verwendet die Steintasten um die jeweils nächste Messung zu starten und euch die Messwerte nacheinander anzeigen zu lassen. Das Einbauen von Signaltönen hilft, das Ende einer Messreihe zu erkennen. Ihr bekommt zusätzlich eine Probe mit einem euch unbekanntem Mischungsverhältnis.	20 min
7. Rechnet eure Messwerte mithilfe des Lambert-Beerschen Gesetzes in Extinktionswerte um, subtrahiert den Wert der Leerprobe und stellt die Endergebnisse grafisch dar. Legt eine Ausgleichsgerade durch eure Messwerte und bestimmt deren Steigung. Berechnet das Mischungsverhältnis eurer zusätzlichen Probe und diskutiert eure Ergebnisse.	15 min

Tab. 5: Unterrichtseinheit „Photometrie“: Arbeitsaufgaben und Durchführungsdauer

zugelassen und somit in der Handhabung unbedenklich. Er eignet sich für dieses Experiment, da sein Absorptionsmaximum bei 608 nm liegt und so das rote Licht des EV3-Lichtensors stark absorbiert. Der Farbstoff ist in vielen Formen erhältlich. Im Rahmen dieses Beispiels wurde eine Tablette Heitmann IRIS-Eierfarbe in 250 ml heißem Wasser gelöst und daraus eine Verdünnungsreihe angesetzt. Idealerweise sind Carbonsäuren schon aus dem Chemieunterricht bekannt, damit ein geeignetes Strukturmodell zur Deutung der Lichtabsorption erarbeitet werden kann. Dieses Experiment könnte jedoch auch auf rein phänomenologischer Ebene durchgeführt werden, ohne dass der Aufbau und die Ursache für die Lichtabsorption von Indigocarmin im Detail erläutert werden. Der Zusammenhang von Informationen und Daten sollte als Dimension der Informatik thematisiert werden. Für den Umgang mit Variablen und Arrays in der EV3-Software ist ggf. eine kurze Einführung nötig. Lichtsensoren und Steintasten sind hingegen durch erklärte Wahlmöglichkeiten relativ verständlich implementiert. Unter *Messen* muss einer der Lichtsensoren auf *Stärke des Umgebungslichts* und der auf *Stärke des reflektierten Lichts* eingestellt werden. Hier



Abb. 2: Aufbau des Experiments mittels Stein, zwei Lichtsensoren und Kvettenhalterung fr photometrische Messungen

sollten aus informatischer Sicht Sensorik, Möglichkeiten der Informationsaufnahme und die Funktionsweise des konkreten Sensors besprochen werden. Die Programmveränderung in Aufgabe 6 ist fakultativ, jedoch lässt sich der Messablauf dadurch beschleunigen und der Umgang mit einer weiteren Variable und verschachtelten Schleifen üben. Das Lambert-Beersche Gesetz und das Umrechnen der Messwerte sollte erläutert werden, da das Rechnen mit Leerprobe und Logarithmus ggf. Schwierigkeiten bereitet. Die Steigung der Messwerte war in einem Kontrollversuch annähernd linear bis zu einem Verhältnis von 1:3. Wenn die Verwendung eines Tabellenkalkulationsprogramms bereits geübt wurde, kann die Steigung der Ausgleichsgeraden natürlich auch von dem Programm bestimmt werden. Über die Geradengleichung kann nun das Mischungsverhältnis der zusätzlich ausgeteilten Proben ermittelt werden.

4 Zusammenfassung und Ausblick

Frei verfügbare, deutschsprachige Unterrichtseinheiten, die als Ziel haben, Informatik fächerübergreifend mit den MNT-Fächern zu unterrichten sowie die konkrete Einbettung in Rahmenlehrpläne, sind rar. Neben den präsentierten Unterrichtseinheiten wurden auch Einheiten zu rationalen Zahlen, ebenen Figuren (Dreiecken und Vierecken), Steuerung und Regelung (Homöostase durch negative Rückkopplung) der thermischen Zustandsgleichung idealer Gase, gleichförmiger Bewegung und gedämpften mechanischen Schwingungen ausgearbeitet. Ob sich diese Unterrichtseinheiten positiv auf die Motivation und die Kompetenzentwicklung von SuS auswirken, ist zu überprüfen. Sie bieten in jedem Fall die Möglichkeit, erste Pilot-Studien durchzuführen und werden interessierten LehrerInnen als OER frei zur Verfügung gestellt. Zur Überprüfung der Einsetzbarkeit der vorgeschlagenen Aufgaben sollten diese zunächst in kleinen Gruppen z. B. an Projekttagen mit SchlerInnen getestet und anschließend ihr Einsatz im Regelunterricht erprobt werden. Im Frhling 2017 wurde das Beispiel zur Biologie (Abschnitt 3.1) mit 7 SchlerInnen der 9. Klasse durchgeführt. Diese gaben anschließend an, dass sie dem Kurs gut folgen konnten und sich wünschen würden, diese Inhalte mit Robotern in der Schule zu behandeln.

Unser Dank gilt dem Humboldt-ProMINT-Kolleg für anregende Gespräche und inhaltliche Unterstützung bei der Entwicklung der Unterrichtseinheiten.

Literaturverzeichnis

- [BB14] Banks, Frank; Barlex, David: Teaching STEM in the Secondary School. Taylor & Francis Ltd, 2014.
- [Br84] Braitenberg, Valerio: Vehicles: Experiments in Synthetic Psychology. MIT University Press Group Ltd, 1984.
- [Ge08] Gesellschaft für Informatik e.V.: Grundsätze und Standards für die Informatik in der Schule. https://www.gi.de/fileadmin/redaktion/empfehlungen/Bildungsstandards_2008.pdf. – Zugriffsdatum: 31.12.16, 2008.
- [KMa] KMK: Bildungsstandards im Fach Biologie für den Mittleren Schulabschluss (Beschluss der Kultusministerkonferenz vom 16.12.2004).
- [KMb] KMK: Bildungsstandards im Fach Chemie für den Mittleren Schulabschluss (Beschluss der Kultusministerkonferenz vom 16.12.2004).
- [KMc] KMK: Bildungsstandards im Fach Mathematik für den Mittleren Schulabschluss (Beschluss der Kultusministerkonferenz vom 04.12.2003).
- [KMd] KMK: Bildungsstandards im Fach Physik für den Mittleren Schulabschluss (Beschluss der Kultusministerkonferenz vom 16.12.2004).
- [KM16] KMK: Bildung in der digitalen Welt. Strategie der Kultusministerkonferenz. https://www.kmk.org/fileadmin/Dateien/pdf/PresseUndAktuelles/2016/Bildung_digitale.-Welt.Webversion.pdf – Zugriffsdatum: 31.12.16, 2016.
- [KWB10] Kocanda, Martin; Wilke, Bryn M.; Ballantine, David S.: Using LEGO MINDSTORMS NXT™ Robotics Kits as a Spectrophotometric Instrument. International Journal on Smart Sensing and Intelligent Systems, 3(3), 2010.
- [NG13] NGSS Lead States: Next Generation Science Standards. <http://nextgenscience.org>. – Zugriffsdatum: 31.12.16, 2013.
- [OE99] OECD: Measuring Student Knowledge and Skills – A New Framework for Assessment. <http://www.oecd.org/edu/school/programmeforinternationalstudentassessmentpisa/33693997.pdf> – Zugriffsdatum: 31.12.16, 1999.
- [OI04] O’Sullivan, Dan; Igoe, Tom: Physical Computing: sensing and controlling the physical world with computers. Course Technology Press, 2004.
- [Se06] Senatsverwaltung für Bildung, Sport und Jugend Berlin: Rahmenlehrplan für die Sekundarstufe I. Jahrgangsstufe 7–10. ITG Informatik als Wahlpflichtfach. 2006.
- [SP15] Schulz, Sandra; Pinkwart, Niels: Physical Computing in STEM Education. In: Proceedings of the Workshop in Primary and Secondary Computing Education. WiPSCE ’15, ACM, New York, NY, USA, S. 134–135, 2015.
- [Th16] The LEGO Group: LEGO MINDSTORMS Education EV3; Aufgaben für den MINT-Unterricht. https://le-www-live-s.legocdn.com/sc/media/files/curriculum/le_ev3_aufgaben_fuer_unterricht-30d0cec3ed3ce82fe35811eed0b42a2e.pdf. – Zugriffsdatum: 31.12.16, 2016.
- [VD07] VDI: Bildungsstandards Technik für den Mittleren Schulabschluss. https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/sk_dateien/V-DI%20Bildungsstandards.2007.pdf. – Zugriffsdatum: 31.12.16, 2007.
- [Ve16] Vernier Software & Technology: Vernier Engineering Projects with LEGO MINDSTORMS Education EV3. <http://www.vernier.com/products/books/ep-ev3/>. – Zugriffsdatum: 31.12.16, 2016.

Ein Framework zur Einordnung programmierbarer Baukästen in interdisziplinäre Bildungskontexte

Eva-Sophie Katterfeldt¹, Nadine Dittert²

Abstract: Aktuell gibt es eine große Anzahl verschiedener programmierbarer Baukästen mit Mikrocontroller für den Bildungsbereich und ständig kommen neue hinzu – das jüngste Beispiel ist Calliope. Sie ermöglichen es nicht nur, informatische und elektrotechnische Themen in kreativem Umgang zu vermitteln, sondern bieten im Sinne einer umfassenden digitalen Bildung Anknüpfungspunkte an eine Vielzahl von Schulfächern. In unserer langjährigen Erfahrung mit dem Einsatz solcher Kits in informellen Bildungskontexten mit diversen interdisziplinären Themenschwerpunkten hat sich gezeigt, dass die jeweiligen Kit-Designs unterschiedliche Eignung aufweisen. Wir präsentieren ein Framework mit Taxonomie, das Lehrende in formellen und informellen Bildungskontexten bei der Wahl passender Baukästen für ihren Anwendungsfall leitet.

Keywords: Construction Kits; programmierbare Baukästen; Framework; Interdisziplinarität.

1 Einleitung und Hintergrund

Programmierbare Baukästen kamen in den 1990er Jahren am MIT auf. Es handelt sich um programmierbare Mikrocontroller für junge Anwender*innen, die mit Sensoren und Aktuatoren erweitert werden. Mittlerweile gibt es eine unglaubliche Vielzahl solcher ‘Kits’ die verschiedene Designs aufweisen [B115]. Um eine Orientierungshilfe ihrer Eignung für interdisziplinäre Projekte zu bieten, haben wir ein Framework entwickelt.

2 Framework und Taxonomie

Das Framework wurde in einem Diskurs wöchentlicher Diskussionen über einen Zeitraum von drei Monaten anhand langjähriger praktischer Erfahrungen mit programmierbaren Baukästen (‘Construction Kits’) mit jungen Menschen im Alter von 9-18 Jahren [Ka15] sowie Literaturrecherchen entwickelt. Gegenstand dieses Artikels ist eine Taxonomie innerhalb des Frameworks, die Merkmale von Kits auf interdisziplinäre Anwendungen bezieht: ‘Welche Merkmale programmierbarer Baukästen sind für bestimmte Anwendungsgebiete relevant?’ Für die ersten ‘programmable bricks’ wurden Projektideen in verschiedenen Anwendungsbereichen vorgeschlagen: das Tracking des eigenen Körpers, die Überwachung von Räumen, das Kreieren ‘intelligenter’ Gegenstände, naturwissenschaftliche Experimente oder das Erschaffen autonomer Wesen [Re96]. Zwar

¹ Universität Bremen, FB3/dimeb, Bibliothekstr. 5, 28359 Bremen, evak@tzi.de

² Universität Bremen, FB3/dimeb, Bibliothekstr. 5, 28359 Bremen, ndittert@tzi.de

sind bei aktuellen Kits neue Möglichkeiten hinzugekommen (z.B. IoT), die genannten sind aber nach wie vor gültig, wie Literaturrecherchen belegen [Pa17]. Basierend darauf wurden folgende Anwendungsfälle für die Taxonomie festgelegt:

- *Geschichtenerzählen*: Projekte, bei denen Artefakte (auch als autonome Wesen) am Rollenspiel beteiligt sind, z.B. einer interaktiven Theateraufführung.
- *Spieldesign*: Entwicklung von Computerspielen, die mit einem Kit (oder Artefakt, in das dies eingebettet ist) als Eingabegerät gesteuert werden.
- *'Intelligente' Gegenstände*: Erfinden und Gestaltung intelligenter Objekte, z.B. durch 'Hacken' bestehender oder Basteln neuartiger Artefakte.
- *Autonome Wesen*: Roboter oder Fahrzeuge, die sich autonom bewegen können.
- *Erfassung körperlicher Aktivität*: 'Tracking' von Bewegungen und Aktivitäten des menschlichen Körpers. Erfordert das Verbinden von Komponenten mit dem Körper.
- *Naturwissenschaftliche Experimente*: Das Erstellen von Messgeräten, die Sensordaten z.B. von Temperatur, Luft, Wasserqualität erfassen und ggf. auswerten.

Diesen Anwendungskategorien werden Merkmale zugeordnet, die erforderlich sind, um entsprechende Projekte möglichst erfolgreich umzusetzen, und die sich auf äußerliche Designmerkmale, technische Schnittstellen und Programmiermöglichkeiten beziehen.

- *Alleinstehend*: Das Kit kann ohne dauerhaften Anschluss an einen Rechner verwendet werden. Wichtig für Projekte, bei denen das Artefakt herumgetragen werden soll. Bspw. nicht erfüllt von LEGO Wedo1.0, im Gegensatz zu Wedo2.0 [LW17].
- *All-in-one*: Sensoren und Aktuatoren sind auf dem Mainboard integriert, so dass es ohne Anschluss weiterer Komponenten einsatzbereit ist. Z.B. [Ca17], [En17].
- *Möglichkeit der Einbettung*: Inwiefern das Kit oder einzelne Komponenten in andere physische Objekte integrierbar sind, z.B. durch flexible Kabellängen, Anschlüsse für nicht-technische Materialien [LM17, PC17] oder geringe Größe ([AB17] Nano).
- *Wearable*: Unterkategorie von 'Einbettung'. Erfüllt von Kits, die sich mit dem Körper verbinden lassen z.B. über Kleidung [BE08].
- *Sensorenvielfalt*: Verfügbarkeit mehrerer verschiedener Sensortypen, die Projektvielfalt und -komplexität unterstützen. Kits mit Breadboards [AB17] oder [AT17] unterstützen durch Erweiterbarkeit viele, LEGO Wedo2.0 nur zwei Sensortypen.
- *Motorik*: Die Anschlussfähigkeit von Motoren, die das Artefakt beweglich machen, z.B. [LW17, LM17]. [AB17] erfordern i.d.R. Erweiterungsshields, die sich auf Einbettung und Tragbarkeit negativ auswirken.
- *Zahlreiche I/O-Schnittstellen*: Relevant für den Anschluss vieler externer Sensoren, Aktuatoren oder zusätzlicher Shields. Z.B. hat [AB17] Mega viele, [Ca17] wenige.
- *USB-Client*: Das Kit wird als Eingabemedium von einem Rechner erkannt, um z.B.

Maus und Tastatur zu ersetzen, z.B. [MM17]. Relevant für Game-Controller-Bau.

- *Kabellose Vernetzbarkeit*: Kits können miteinander oder mit anderen Geräten drahtlos kommunizieren. Relevant für Theateraufführungen oder IoT, z.B. [LM17] EV3.
- *Programmierbarkeit*: Inwiefern Kits programmierbar sind. Z.B. sind [LB17] originär nicht programmierbar, daher ist ihr Verhalten begrenzt. Erhöht sich bei Kits, die umfangreiche textuelle Programmiersprachen (z.B. Arduino IDE) unterstützen.
- *Datenspeicherung*: Das Board kann größere Datenmengen speichern. Relevant u.a. bei naturwissenschaftlichen Experimenten. Z.B. [En17] mit SD-Kartensteckplatz.

Tabelle 1 ordnet relevante Kit-Merkmale den Anwendungsfällen zu.

Merkmal erWünscht o. erForderlich für Anwendungsfall	Ge-schich-tenerez.	Spiele-sign	Intel. Objekte	Auton. Wesen	Erf. körperl. Akt.	naturw. Exp.
Alleinstehend	F		F	W	F	W
All-in-one		W				
Einbettung			F		F	W
Wearable	W				F	
USB-client		F				
Sensorvielfalt	W		W	W	F	F
Motorik	F			F		
Zahlreiche I/O			W		W	W
kabellose Konnektivität	W		W		W	W
Programmierbarkeit	W	W	W	W	F	F
Datenspeicherung						W

Tab. 1: Taxonomie der Objekt-Perspektive: Kit-Eigenschaften nach Anwendungsfall.

3 Anwendung und Diskussion des Frameworks

Unser Framework konzentriert sich darauf, Kits mit Anwendungsgebieten in Verbindung zu bringen, für die sie sich anhand ihrer baulichen Merkmale am ehesten eignen. Für ein Projekt können mehrere Anwendungsfälle der Taxonomie relevant sein. Z.B. ein Theaterstück mit Robotern (*Geschichtenerzählen* und *Autonome Wesen*) oder Messung und Auswertung eigener Bewegungen (*Naturwiss. Experimente* und *Erfassung körperlicher Aktivität*). Ebenso kann die Taxonomie genutzt werden, um geeignete Anwendungen für vorrätige Kits zu identifizieren. Mitunter ist kein klares Ergebnis hinsichtlich der Eignung eines Kit-Typs zu erwarten und weitere kontextuelle Aspekte sind abzuwägen (z.B. Vorkenntnisse). Beispielsweise hängt eine ausreichende Vielfalt an Sensoren ab von der

angestrebten Artefaktkomplexität, vom Vorwissen der Beteiligten und Projektdauer.

4 Fazit und Ausblick

Dieser Beitrag liefert ein Framework, das Merkmale programmierbarer Baukästen typischen interdisziplinären Anwendungsfällen zuordnet und Orientierung bei der Wahl passender Kits im Bildungsbereich bietet. Zukünftig wollen wir das Framework um Perspektiven auf beteiligte Akteur*innen und Lernumgebung und -kontext erweitern.

5 Danksagung

Diese Forschung war Teil des Projekts PELARS innerhalb des 7. Rahmenforschungsprogramms der Europäischen Union unter Förderkennzeichen 619738.

Literaturverzeichnis

- [AB17] Arduino Boards, <https://www.arduino.cc/en/Main/Products>, Stand: 26.05.2017.
- [AT17] Arduino TinkerKit, <http://arduino.org/products/tinkerit>, Stand: 11.02.2017.
- [Bl15] Blikstein, P.: Computationally Enhanced Toolkits for Children: Historical Review and a Framework for Future Design. *Foundations and Trends® in HCI*. 9, 1, 1–68, 2015.
- [BE08] Buechley, L., Eisenberg, M.: The LilyPad Arduino: Toward Wearable Engineering for Everyone. *IEEE Pervasive Computing*. 7, 2, 12–15, 2008.
- [Ca17] Calliope, <http://www.calliope.cc>, Stand: 11.02.2017.
- [En17] Engduino, <http://www.engduino.org>, Stand: 11.02.2017.
- [Ka15] Katterfeldt, E.-S. et al.: Designing digital fabrication learning environments for Bildung: Implications from ten years of physical computing workshops. *International Journal of Child-Computer Interaction*. 5, 3–10, 2015.
- [LB17] LittleBits. <https://littlebits.cc>, Stand: 26.05.2017.
- [LM17] LEGO Mindstorms, <https://www.lego.com/de-de/mindstorms>, Stand: 11.02.2017.
- [LW17] LEGO WeDo, <https://education.lego.com/de-de/grundschule/shop>, Stand: 11.02.2017.
- [MM17] MakeyMakey, <http://makeymakey.com>, Stand: 11.02.2017.1.
- [Pa17] Papavlasopoulou, S. et al.: Empirical studies on the Maker Movement, a promising approach to learning: A literature review. *Entertainment Computing*. 18, 57–78, 2017.
- [PC17] PicoCricket, <http://picocricket.com>, Stand: 11.02.2017.
- [Re96] Resnick, M. et al.: Programmable Bricks. *IBM Sys. Journal*. 35, 3.4, 443–452, 1996.

Das *EduFab*-Kit – Ein modularer FabLab-Baukasten für Lehrer*innen

Nadine Dittert¹, Eva-Sophie Katterfeldt²

Abstract: Das digitale Herstellen von Produkten – oder *Making* – ist in den Bildungsbereich vorgedrungen. Die Idee des konstruktiven Arbeitens mit digitalen Medien ist nicht neu, erhält aber durch FabLabs und Makerspaces und dem damit geschaffenen Zugang zu 3D-Druckern, Laser-Cuttern, etc. neuen Nährboden für das Lernen. Neue Anwendungsmöglichkeiten allein genügen jedoch nicht: es braucht Konzepte, Materialien und konkrete Ideen, wie Technologie Teil eines Lernprozesses werden kann. Wir präsentieren das *EduFab* - Kit, das Lernaktivitäten im FabLab für Anfänger*innen unterstützt. Das Kit hilft Lehrer*innen dabei, Lernumgebungen zu schaffen, in denen Schüler*innen ausgehend von ihren eigenen Ideen Produkte mit dem 3D-Drucker, dem Laser-Cutter, Mikrocontrollern o.ä. entwickeln. Das Kit besteht aus Materialien in einem modularen Koffer, einem generellen Konzept zur Arbeit mit jungen Menschen im FabLab im Bildungskontext und Beispielworkshops, die mit dem Kit durchgeführt werden können. Weiterhin beinhaltet das Kit Anleitungen und Listen für zusätzlich notwendiges Material und Maschinen des FabLabs.

Keywords: FabLab; Kit; außerschulischer Lernort; Informatik; Digitale Medien; Lernen; Schüler*innen; Konstruktivismus.

1 Einleitung

Fabrication Laboratories (FabLabs) ermöglichen Menschen verschiedenen Hintergrunds den Zugang zu High-Tech-Maschinen, die zuvor einem eher begrenzten Teil der Bevölkerung vorbehalten war. Technologien wie Mikrocontroller, 3D-Drucker, 3D-Scanner, Laser-Cutter und CNC-Fräsen bieten grundlegend neue Möglichkeiten zur kreativen Gestaltung von (digital entwickelten) Produkten. Genutzt wird dies von Menschen, die aus verschiedenen persönlichen und auch beruflichen Interessen heraus Dinge herstellen. Dieses Herstellen – Englisch *making* – rückt nicht ohne Grund mehr und mehr in den Mittelpunkt von Lernprozessen: es macht digitale Modelle „be-greifbar“.

2 Hintergrund

Der Begriff „Be-greifbarkeit“ verdeutlicht das Zusammenspiel von Anfassen und Verstehen, von Fühlen und Denken [RS12], was der aktive Umgang mit FabLab-

¹ Universität Bremen, FB3/dimeb, Bibliothekstr. 5, 28359 Bremen, ndittert@tzi.de

² Universität Bremen, FB3/dimeb, Bibliothekstr. 5, 28359 Bremen, evak@tzi.de

Technologien grundsätzlich in sich birgt. Im FabLab steht das Erschaffen von Objekten im Mittelpunkt. In einem konstruktionistischen Lernprozess ist ein solches Objekt der sogenannte „Gegenstand-mit-dem-man-denkt“ – es ist Ausdruck gedanklicher Modelle, die reflektiert und angepasst werden können [Pa80]. Nicht jede Konstruktionstätigkeit im FabLab bedeutet gleichzeitig, dass konstruktionistisches Lernen stattfindet. Neben der Notwendigkeit der Reflexion des Prozesses verlangt dieses Lernparadigma einen wesentlichen Aspekt: die persönliche Bedeutsamkeit des erschaffenen Produkts.

Aus diesem Blickwinkel betrachtet das Projekt *EduFab* das FabLab als Lernort für Schüler*innen. Die Möglichkeiten, die die Maschinen an sich bieten, müssen didaktisch aufbereitet werden, um einen konstruktionistischen Lernprozess zu ermöglichen. Die Schüler*innen müssen dabei begleitet werden, aus einer Idee mit der vorhandenen Technologie ein Produkt zu entwickeln. Gleichzeitig soll die persönliche Bedeutsamkeit berücksichtigt werden. Dazu soll der Prozess um die Ideenfindung und die Präsentation des Produkts erweitert werden. Um ein FabLab – oder ähnlich ein Makerspace oder Hackerspace – zu einer Lernumgebung nach diesen Kriterien zu machen, braucht es passende Konzepte und Materialien.

Für diesen Zweck haben wir das *EduFab*-Kit entwickelt. Es ist eine Zusammenstellung von Materialien, Werkzeugen und Listen, die die Nutzung des FabLabs als Lernort für und mit jungen Menschen fördert. In einem didaktischen Konzept werden Anfänger*innen zu Aktivitäten angeregt und dazu befähigt, zum Maker oder zur Makerin zu werden. In unterschiedlichen von uns ausgearbeiteten Szenarien geht es darum, dass Schüler*innen fantasieren, explorieren, kreieren, modellieren, programmieren, drucken und cutten. Schlussendlich halten sie ein Produkt in der Hand, das ausgehend von ihrer Lebenswelt und nach ihren Ideen entsteht.

Als Arbeitsgruppe „Digitale Medien in der Bildung“ (dimeb) blicken wir auf langjährige Forschungs- und Erfahrungserfahrung zum Einsatz Digitaler Medien im Bildungskontext zurück. Das Aufkommen von FabLabs und deren Nutzen für das Lernen ist in den letzten Jahren ins Zentrum unserer Forschung gerückt. Dabei haben wir unsere Erkenntnisse, die vor allem aus dem Bereich des Umgangs mit Mikrocontroller-Baukästen in Lernkontexten stammen, auf die neuen Technologien angewandt und in diesem erweiterten Bereich geforscht und entwickelt. Ein Ergebnis ist das hier dargestellte *EduFab*-Construction Kit.

3 Das *EduFab*-Kit

Das *EduFab*-Construction Kit (kurz: *EduFab*-Kit) wurde entwickelt, um Schüler*innen die FabLab-Umgebung und deren Technologien konkret anhand von Beispielen erfahrbar zu machen. Eingebettet in ein didaktisches Konzept können die Technologien erkundet und mit ihnen experimentiert werden, um zu Grunde liegende Funktionsweisen zu „be-greifen“. Das *EduFab*-Kit soll Lehrer*innen Wege eines geeigneten Zugangs zur Arbeit mit Kindern und Jugendlichen im FabLab zeigen und zur Vermittlung und

produktiven Nutzung von FabLab-Technologien für Kinder und Jugendliche anregen. Es richtet sich zunächst an Menschen ohne Vorerfahrung mit FabLab-Technologien und soll dabei helfen, deren Neugier zu wecken. Es kann jedoch auch erweitert und darüber hinaus genutzt werden.

Das *EduFab*-Construction Kit besteht aus verschiedenen Teilen:

- einem (Material-) Koffer mit „Tools“ zum Arbeiten,
- einem (Workshop-) Konzept als Vorschlag für die Arbeit mit jungen Menschen im FabLab,
- ausgearbeiteten Beispielworkshops, bei denen Anfänger*innen im FabLab mit verschiedenen Technologien in Berührung kommen sowie
- Anleitungen, wie einer Bauanleitung zum Koffer, und Listen über weitere benötigte (Arbeits-) Materialien, Software und Anregungen für weitere Arbeiten im FabLab.

Zusätzlich wird zu diesem Kit ein Materiallager benötigt, in dem beispielsweise Filament für den 3D-Drucker sowie Holz-, Acryl- oder Pappplatten für die Arbeit mit dem Laser-Cutter in ausreichenden Mengen vorhanden sind. Dies ist meist in FabLabs vorzufinden. Vorschläge für eine Mindestausstattung an Material, das für die spezifischen Workshops zur Verfügung stehen sollte, sind in den Anleitungen enthalten und dienen als Grundlage für Absprachen mit dem FabLab, in dem gearbeitet wird.

Das *EduFab*-Construction Kit selbst folgt der Idee des „Learning-by-doing“. Lehrer*innen werden selbst aktiv, bevor sie mit den Schüler*innen arbeiten. Ihre Arbeit beginnt „Hands on“ mit dem Bau eines eigenen *EduFab*-Koffers. Dieser liegt als digitales Modell zur eigenständigen Produktion vor. Der Inhalt des Koffers steht (zunächst) in Form von Materiallisten zur Verfügung, Handreichungen und Materialien stehen in Papierform oder als druckfertige Dateien bereit.

Der Koffer „wächst“ mit den Erfahrungen der Lehrenden – je mehr Workshops durchgeführt wurden, umso mehr Ebenen existieren für den Koffer. Seine Modularität lädt zusätzlich dazu ein, eigene Workshops zu entwickeln und dafür eigene Ebenen zu erstellen.

3.1 Der *EduFab*-Koffer

Der *EduFab*-Koffer ist ein im FabLab mit dem Laser-Cutter hergestellter modular gestalteter Kasten (siehe Abbildung 1). Er enthält die „Tools“, die für die Arbeit im FabLab benötigt werden. Unter anderem ist damit seine eigene Produktion bzw. die weiterer Koffer möglich – der Koffer selbst ist ein autoreplikatives System.

Der Grundgedanke ist dabei, mit dem Bauen des Koffers die erste eigene praktische Arbeit im FabLab umzusetzen, sozusagen ein „Hello World“ mit dem Laser-Cutter. Bei

der Konzeption des Koffers wurde darauf geachtet, eine möglichst vielfältige Bandbreite an Arbeitstechniken mit dem Laser-Cutter abzudecken (Schnitt und Gravur, Steckmechanismen, Flexibilität) um eine aktive Auseinandersetzung mit diesen Techniken anzuregen.



Abb. 1: Der *EduFab*-Materialkoffer

Der *EduFab*-Koffer besteht aus verschiedenen Ebenen, die aufeinander stapelbar sind. Diese Bauweise ermöglicht es, individuell Inhalte zu verändern, auf verschiedene Bedürfnisse anzupassen und sie selbst zu erweitern.

Die unterste Ebene beinhaltet das Basiskit, das grundsätzlich für die Arbeit im FabLab hilfreich ist und vorwiegend für die Anfertigung des Koffers selbst notwendig ist. Sie ist, im Gegensatz zu den weiteren Ebenen, ständiger Bestandteil des Materialkoffers und personalisierbar. Die Grundebene besteht aus einem Sortierkasten mit einem nach oben herausragenden Griff und einem Deckel. Die Größe wird durch ein darunter liegendes Fach bestimmt, in das Papier der Größe DIN A4 passt.

Zusätzlich gibt es für verschiedene Workshops weitere Ebenen, die das spezifische Material für ein konkretes Workshopszenario enthalten (siehe Abbildung 2). Diese Sortierkästen ähneln der Grundebene, haben jedoch statt des Griffs eine Öffnung und

können so auf die Grundebene aufgesetzt werden. Sie enthalten herausnehmbare Elemente, so dass die Größe der jeweiligen Fächer anpassbar ist. Auf die oberste Ebene wird der Deckel gepackt und der Kasten lässt sich als geschlossener Materialkoffer transportieren.

Auf diese Weise ist der *EduFab*-Materialkoffer modular gehalten. Für einen speziellen Workshop werden Grundebene und Workshop-spezifische Ebene zusammengesteckt. Gleichzeitig ist der Materialkoffer flexibel erweiterbar: Nutzende sind eingeladen, Teile des bestehenden Systems auf eigene Bedürfnisse anzupassen oder auch weitere Ebenen und Materialien zu entwickeln und mit der Community der Maker*innen und Fabber*innen zu teilen.



Abbildung 2: Die Schmuckebene des *EduFab*-Materialkoffers

Aktuell existieren fünf verschiedene Ebenen, die den diversen Möglichkeiten der Digitalen Fabrikation im FabLab – orientiert an den Geräten des FabLab Bremen – zugeordnet sind und in dem Dokument „Der *EduFab*-Koffer :: Der Inhalt“³ näher erläutert werden. Für jede dieser Ebenen stehen zusätzlich zu den „Tools“ Materialien zur Erläuterung bzw. zum Umgang im Workshop zur Verfügung.

³ http://www.dimeb.de/edufab/wordpress/wp-content/uploads/2016/11/edufab_koffer_inhalt.pdf

3.2 Das Workshopkonzept

Das Workshopkonzept dient als Vorschlag für den Einstieg in die Arbeit mit jungen Menschen im FabLab. Es beschreibt, wie die digitale Fabrikation mittels FabLab-Technologien (z.B. 3D-Drucker, Laser-Cutter, Plotter, Arduino) in einem Workshop mit Kindern und Jugendlichen didaktisch begleitet werden kann. Es ist angelehnt an das TechKreativ-Konzept [Di12], das nun seit mehr als 10 Jahren angewandt und weiterentwickelt wird, um es (jungen) Menschen ohne Vorkenntnissen zu ermöglichen, selbst Technologie zu konstruieren und gleichzeitig einen Bildungsbeitrag zu leisten [Ka15]. In diesem Rahmen wird Informatik „gemacht“ – von einer Problemstellung über eine praktische Lösung (inklusive eines Programmierteils) zur Präsentation. Die Kinder und Jugendlichen werden dazu in ihrer Erlebniswelt „abgeholt“ und dabei begleitet, eigene Ideen umzusetzen und am Ende zu präsentieren. Der Prozess verläuft in fünf Phasen, die im Folgenden beschrieben werden.

Ziel der ersten Phase, der *Fantasiephase*, ist eine freie Idee, die aus der Lebenswelt der Teilnehmenden kommt. Sie wird mittels Kreativmethoden, beispielsweise durch ein Brainstorming, generiert. Die Teilnehmenden sollen sich mit der Aufgabe identifizieren und ihre eigenen Ideen einbringen können, da die persönliche Bedeutsamkeit eine wichtige Grundlage für das Lernen im Workshop darstellt.

Im Anschluss an die Fantasiephase wird die Technologie vorgestellt (*Phase 2: Technologieeinführung*). In kleinen Gruppen werden den Teilnehmenden an Stationen die verschiedenen zur Verfügung stehenden Technologien erklärt. Dies geschieht so praktisch wie möglich, ohne jedoch konkrete Beispiele zu benennen, die im Workshop nachgebaut werden können. Eine kleine Hands-on-Aufgabe, bei der die Teilnehmenden erfolgreich aktiv werden, beendet die Technologieeinführung. An dieser Stelle wird auch das zu verwendende Material gezeigt.

Das Wissen über Technologie und Material und die Ergebnisse der Fantasiephase dienen als Ausgangspunkt zur Entwicklung konkreter Ideen (*Phase 3: Ideenfindung*). Aufgabe der Tutorin oder des Tutors ist es, gemeinsam mit den Teilnehmenden eine im Rahmen des Workshops umsetzbare Idee pro Kleingruppe zu entwickeln. Diese hat am Ende dieser Phase eine wahrnehmbare Form, z.B. als Beschreibung oder Zeichnung.

In der vierten Phase (*Konzeptentwicklung, Konstruktion und Programmierung bzw. Modellierung*) beginnt nun die genauere Konzeption des Objekts sowie dessen Umsetzung. Die Teilnehmenden setzen aktiv ihre Ideen um und werden dabei von den Tutor*innen unterstützt, jedoch nicht instruiert. Am Ende der vierten Phase steht der „finale Prototyp“ – das Produkt.

Zuletzt wird das Produkt präsentiert (*Phase 5: Präsentation*) und durch die Verbalisierung reflektiert. Der Workshop endet mit einem positiven Erlebnis.

Im Laufe des Workshops ist es wichtig, kontinuierlich Bezüge der aktuellen Arbeit zur Erlebniswelt der Teilnehmenden, zum Alltag und zur Welt herzustellen. Dabei wird auf

ähnliche Anwendungen sowie auf Anwendungsbereiche wie Forschung, Medizin, Kunst und Hobby eingegangen.

3.3 Drei ausgearbeitete Beispielworkshops

Im Rahmen des Projekts sind drei beispielhafte Workshops, bei denen Anfänger*innen im Alter von 13 bis 19 Jahren im FabLab mit verschiedenen Technologien in Berührung kommen, entstanden. Mit diesen konkreten Anleitungen können Lehrer*innen oder Tutor*innen die Workshops durchführen⁴. Dabei wird das zuvor beschriebene Konzept angewandt. Das notwendige Material befindet sich in der dazu gehörenden Ebene des Koffers.

Der Kurzworkshop „Stoffdrucke programmieren“ bietet Anfänger*innen einen praktischen Einstieg in textuelle Programmierung mit Processing und setzt das Programmierte künstlerisch um. In drei Stunden programmieren die Teilnehmenden aus einfachen Formen eine Figur oder ein Muster, die sie mit dem Plotter ausschneiden und dann auf einen Stoffbeutel aufbügeln können. Inhaltlich sehr ähnlich ist der Kurzworkshop „Programmierte Schmuckstücke“ in dem ebenso ein Muster in Processing programmiert wird. Hierbei entstehen jedoch Muster für Kettenanhänger oder Ohrringe, die mit dem Laser-Cutter ausgeschnitten und – wenn gewünscht – graviert werden. In dem Workshop „Analytische Geometrie am Beispiel von Gebäuden“ werden Grundelemente der analytischen Geometrie am Beispiel des Modellbaus von Gebäuden eines zentralen Platzes im Lebensraum der Lernenden erarbeitet. Arbeitsteilig in Kleingruppen werden einzelne Gebäudemodelle erstellt und dann auf dem Modellplatz gemäß ihrer exakten Position aufgestellt.

3.4 Weitere Anleitungen

Als weitere Anleitungen stehen u.a. eine Bauanleitung zum Koffer, die Schnittdatei und der Inhalt des Koffers einschließlich Listen über weitere benötigte (Arbeits-) Materialien online zur Verfügung.

4 Erfahrungen im Umgang mit dem *EduFab*-Kit

Besonders viel Arbeit und Überlegungen sind in die Anpassung des bewährten TechKreativ-Konzeptes auf die „neuen“ Maschinen geflossen. Dabei waren uns zwei Aspekte besonders wichtig: der Anschluss an die Lebenswelt der Teilnehmenden und die Nutzung der Produktionsmaschinen für einen konstruktiven Einblick in die Informatik. Die im Folgenden beschriebenen Erfahrungen beziehen sich insbesondere auf die

⁴ Alle Anleitungen zu den Workshops sowie zum Koffer sind unter <http://www.dimeb.de/edufab/material> zu finden.

Workshops „Stoffdrucke programmieren“ und „Programmierte Schmuckstücke“⁵.

Um an die Interessen der Kinder und Jugendlichen anzuschließen, suchten wir nach Szenarien, in denen Dinge entstehen, die sie im Alltag nutzen können und die gleichzeitig eine persönliche Note zulassen. Die Wahl von Schmuckstücken und Stoffbeuteln hat sich hierbei als geeignet und darüber hinaus als interessantes Thema für diverse Zielgruppen (u.a. Mädchen, Programmierinteressierte, Geflüchtete) erwiesen. Die Teilnehmer*innen waren stolz auf ihre Produkte, deren Nutzung im Alltag sie in den Workshops angekündigt hatten. Persönliche Bezüge zeigten sich in den entworfenen Mustern und Formen: zwei Teilnehmerinnen, die in der Fantasiephase einen Fisch zeichneten, weil sie gerne schwimmen, setzten diesen um, ebenso wie zwei Mädchen, die gerne Eis essen und in der Fantasiephase eine Eiswaffel zeichneten. Beides wurde programmiert und geplottet (siehe Abbildung 3). Zahlreiche weitere umgesetzte Beispiele dieser Art zeigen, dass der Umgang mit dem *EduFab*-Kit einschließlich der Anwendung des Konzeptes die Konstruktion persönlich bedeutsamer Objekte fördert.

Das Ziel, mit dem *EduFab*-Kit einen aktiven Einblick in die Informatik zu geben, wurde umgesetzt, indem einerseits der Prozess der Problemlösung aktiv durchlaufen wird und andererseits darin Formen und Muster programmiert werden, statt sie mit einem Grafikprogramm zu erzeugen. Diese Szenarien sind nur ein kleiner Einblick und zeigen lediglich allererste Programmierschritte. Nichtsdestotrotz wurde durch Programmierung, also durch „dem Computer in einer seiner Sprachen zu sagen, was er tun soll“, ein Objekt erschaffen, statt den Computer als Zeicheninstrument zu nutzen. Die Szenarien sind durchaus erweiterbar, was bereits teilweise erfolgreich getestet wurde.

Das eigenständige Konstruieren eines *EduFab*-Koffers durch Lehrer*innen konnte aus Zeitgründen bisher nicht getestet werden. Ein Testlauf mit zwei Teilnehmerinnen hat jedoch gezeigt, dass es sinnvoll ist, die Konstruktion eines Koffers durch Tutor*innen zu begleiten, die an einigen Stellen hilfreiche Tipps geben können. Für Workshops mit mehreren Lehrer*innen sollten einige Teile des Koffers aus zeitlichen Gründen bereits im Vorfeld gecuttet werden.

Das Design des Koffers fand bisher viel Lob und Zustimmung, ebenso wie die Idee, darin Workshopmaterialien aufzubewahren. In unseren Workshops nutzen wir gerne diese Gelegenheit, uns ist jedoch nicht bekannt, ob Lehrer*innen tatsächlich mit dem entworfenen Koffer und ihrer Klasse ins FabLab gehen.

Ein nicht zu verschweigender Aspekt ist der Kostenaufwand der beschriebenen Aktivitäten. Die Produkte selbst, die Nutzung der Maschinen und das (zusätzlich notwendige) Personal müssen berücksichtigt werden. Im Rahmen des Projekts haben wir teilweise versucht, Kostenfaktoren gering zu halten. Das Stoffdrucke-Szenario gleicht inhaltlich bewusst dem Schmuckszenario, weil ein Plotter wesentlich günstiger zu

⁵ Im Geometrie-Workshop steht das Anwendungsfeld Geometrie im Vordergrund, nicht die Informatik. Die persönliche Bedeutsamkeit ist nicht vergleichbar mit den beiden genannten Workshops. Dennoch wurde bewusst der heimische Markt als Szenario ausgewählt, um einen persönlichen Bezug zuzulassen.

betreiben ist bzw. aufgrund seiner vergleichsweise geringen Anschaffungskosten eventuell sogar Teil der Schulausstattung sein könnte. Nichtsdestotrotz ist ein *EduFab*-Workshop nicht vergleichbar mit einer Klassenraumsituation, in der eine Lehrkraft mit 25 Schüler*innen an Rechnern arbeitet.

```
size(800,800);  
background(255);  
  
triangle(250,250,550,250,400,600);  
ellipse(325,175,150,150);  
ellipse(475,175,150,150);  
arc(400,100,150,150,PI,2*PI);
```

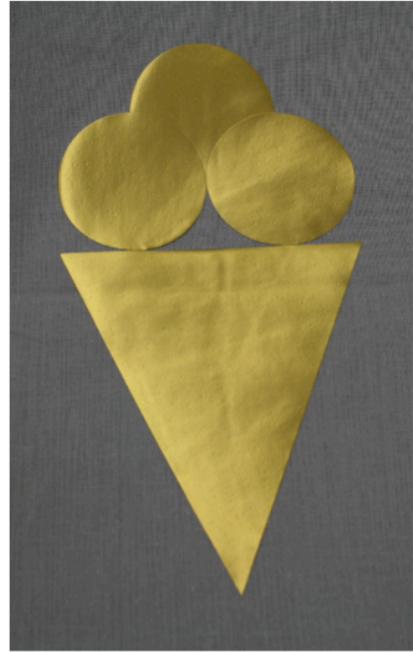
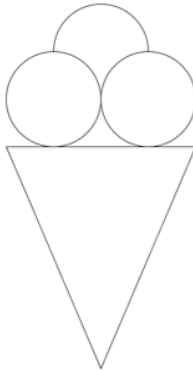
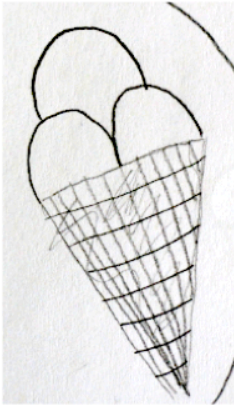


Abb. 3: Skizze, Quellcode, Processing-Ergebnis und Stoffdruck einer Eiswaffel

5 Fazit und Ausblick

Das *EduFab*-Kit bietet Lehrer*innen einen Zugang zum FabLab bzw. zu Technologien des FabLabs und Ideen für Workshops mit jungen Menschen. Die dargestellten Beispiele zeigen, dass es möglich ist, FabLab-Technologien zu nutzen, um jungen Menschen einen praktischen Zugang zur Informatik zu bieten, in dem ein (persönlich bedeutsames) Produkt entwickelt wird.

Am Ende eines *EduFab*-Workshops steht ein Produkt, das i.d.R. den Teilnehmenden am Ende gehört. Dieses Produkt kann getragen und gezeigt werden, wodurch wieder und wieder Anerkennung für die verrichtete Arbeit erfahren wird. Dennoch steht in *EduFab*-Workshops der Prozess im Vordergrund. Es wäre durchaus weniger kompliziert, fertige Designs aus dem Netz zu laden, oder mit Zeichenprogrammen zu modellieren und diese auf Taschen zu drucken. Mit hoher Wahrscheinlichkeit wären diese sogar ästhetischer

als die programmierten Werke, die (insbesondere im vorgeschlagenen Zeitrahmen) einen eher stilistischen Charakter haben. Sie wären nur eben nicht selbst programmiert. Das *EduFab*-Kit hat den Anspruch, Wege und Möglichkeiten zu zeigen, mit heute zugänglichen Produktionsmaschinen Informatik „zu machen“.

Die bisherigen Beispielworkshops sind lediglich Einstiegworkshops, die bezüglich der Informatik das Potential der Maschinen im FabLab nicht ausreizen. Die Szenarien sind erweiterbar. So eignen sich Schleifen zur Erzeugung von sich wiederholenden Mustern, die hervorragende Taschen- oder Schmuckdesigns darstellen können. Für das Entwickeln interaktiver Gadgets befindet sich Mikrocontroller-Material in der SmartFab-Ebene. Schlussendlich sind die bisher vorhandenen Ebenen und Konzepte Vorschläge, die auch als Anregung für eigene Entwicklungen dienen sollen. Aus diesem Grund ist der Koffer modular gehalten. Das allgemeine Workshopkonzept lässt sich in diversen Settings anwenden und darf und soll für eigene Zwecke angepasst und verändert werden.

6 Danksagung

Das Projekt *EduFab* wurde vom Bundesministerium für Bildung und Forschung gefördert. An der Implementierung und dem Design des Koffers war Marvin Lange maßgeblich beteiligt. Es basiert auf einer Arbeit unseres Kollegen Florian Lütkebohmert.

Literaturverzeichnis

- [Di12] Dittert, N. et al.: TechKreativ: Tangible Interfaces in Lernwelten. In Robben, B. & Schelhowe H., eds. Be-greifbare Interaktionen - Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing. Bielefeld: transcript. 2012.
- [Ka15] Katterfeldt, E.-S. et al.: Designing digital fabrication learning environments for Bildung: Implications from ten years of physical computing workshops. International Journal of Child-Computer Interaction. 5, 3–10, 2015.
- [Pa80] Papert, S.: Mindstorms: children, computers, and powerful ideas. Basic Books, Inc. New York, NY, 1980.
- [RS12] Robben, B.; Schelhowe, H.: Be-greifbare Interaktionen - Der allgegenwärtige Computer: Touchscreens, Wearables, Tangibles und Ubiquitous Computing. Transcript, Bielefeld, 2012.

3D-Druck als Motivation für die Einführung in die Programmierung in der Realschule

Oliver Krisch¹, Petra Kastl² und Ralf Romeike³

Abstract: 3D-Drucker gehören derzeit zu den Informatiksystemen, die für die Schüler noch neu sind, Neugier hervorrufen und sich somit gut einsetzen lassen, um sie zu motivieren. Erschöpft sich die Nutzung eines 3D-Druckers allerdings allein im Herunterladen und Ausdrucken vorgefertigter Modelle, haben die Schüler aber wenig über Informatik gelernt. In diesem Beitrag wird eine Unterrichtssequenz von 7 Doppelstunden zum Erlernen grundlegender Programmierkenntnisse vorgestellt, die die Programmierung dreidimensionaler Objekte mithilfe von Turtle-Grafiken in Beetle Blocks zum Ziel hat. Zudem sollte noch eine fächerübergreifende Verbindung zu den im Mathematikunterricht kennengelernten Funktionen geknüpft werden. Als Motivation diente das Erstellen eines 3D-Körpers, der am Ende der Sequenz mit Hilfe eines 3D-Druckers ausgedruckt werden sollte. Die dargestellten Erfahrungen zeigen, dass sich die Schüler von der Idee begeistern lassen, die Verknüpfung mit mathematischen Grundlagen allerdings eine Herausforderung darstellt.

Keywords: 3D-Druck, Realschule, Programmierung, Beetle Blocks

1 Einleitung

Nachdem lange Zeit der wahrnehmbare technische Fortschritt vor allem durch bessere Grafikleistungen bestimmt war, die in der Schule im Bereich der Computergrafik Schülerinnen und Schüler motivieren konnten, sind 3D-Drucker eine jüngere technische Entwicklung, die i.d.R. (noch) nicht im typischen Schülerhaushalt zu finden ist, aber durch erstaunliche Ergebnisse Schüler, Kollegen und Eltern zu faszinieren weiß. An verschiedenen Schulen werden deshalb 3D-Drucker angeschafft, um Schülern die beeindruckenden Möglichkeiten des 3D-Drucks zu eröffnen. In der Praxis beschränkt sich die Verwendung von 3D-Druckern allerdings häufig auf das Konfigurieren des Druckers und das Ausdrucken von aus dem Internet heruntergeladenen vorgefertigten Modellen. Für die Nutzung des fachübergreifenden Potenzials im Informatikunterricht wurden bisher wenig überzeugende Beispiele publiziert. Im Folgenden stellen wir ein Beispiel vor, in dem von Schülern algorithmisch erzeugte 3D-Modelle und deren 3D-Ausdruck als Gegenstand und Motivierung der Einführung in die Programmierung herangezogen werden. Dabei experimentie-

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, oliver.krisch@fau.de

² Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, petra.kastl@fau.de

³ Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, ralf.romeike@fau.de

ren die Schüler mit ihnen aus dem Mathematikunterricht bekannten Konzepten, deren Verwendung in Verbindung mit einfachen algorithmischen Grundstrukturen erst die beeindruckenden Figuren ermöglicht und damit den Anwendungskontext außerhalb der Mathematik eröffnet.

2 Geometrische Formen und algorithmische 3D-Modelle

2.1 Turtlegrafik als Basis für algorithmische 3D-Modelle

Seymour Papert stellte in den 1980er Jahren mit Logo eine Programmierumgebung vor, mit der Kinder geometrische Formen erkunden können, indem sie diese mit einfachen Programmieranweisungen am Bildschirm selbst zeichnen. Papert [Pa82] unterstreicht mit dem von ihm gleichzeitig als Konstruktivismus vorgestellten Ansatz die Rolle des Schaffens konkreter Artefakte, die persönlich bedeutungsvoll sind und anderen gezeigt, erprobt und auch bewundert werden können. Der als Turtle-Grafik bekannte Ansatz liegt auch den folgenden Beispielen zugrunde, allerdings erweitert auf den dreidimensionalen Raum: Die Ausgabe des 3D-Druckers bzw. deren virtuelle Darstellung entspricht dem nachgezeichneten Weg der Schildkröte in Logo oder dem Stift eines Objekts in Scratch.

Viele Gegenstände des alltäglichen Lebens lassen sich aus mathematischen Körpern und Formen aufbauen, welche sich sehr gut als Grundlage für Turtle-Grafiken eignen. Ausgangspunkt in unserem Beispiel war ein Salzstreuer, dessen Form einem Hyperboloid mit der Formel $\frac{x^2}{5,5^2} - \frac{y^2}{2,5^2} = 1$ ähnelt (vgl. Abb. 1) und die zugrundeliegende Idee, Körper aus mathematischen Funktionen zu erstellen und mit einem 3D-Drucker auszudrucken. Zur Umsetzung wurde die blockorientierte Programmiersprache Beetle Blocks [KR12] verwendet.

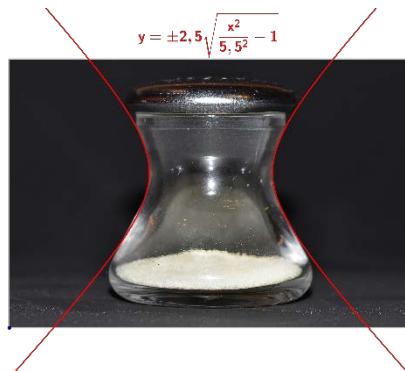


Abb. 1: Salzstreuer – Hyperboloid als Vorlage für 3D-Objekte.

Für den vorliegenden Salzstreuer diente der Kreis als Ausgangsfigur. Die Schülerinnen und Schüler erzeugten die Kreislinie mit Hilfe der Formel $r^2 = x^2 + y^2$. Diese stellten sie zuerst um zu $y = \pm\sqrt{r^2 - x^2}$ und teilten das Zeichnen des Kreises auf in das Zeichnen zweier Halbkreise. Damit konnten die Schülerinnen und Schüler nun einen Zylinder zeichnen, indem sie die Höhe z langsam erhöhten. In einem zweiten Schritt lernten sie dann den Radius der einzelnen Kreise in Abhängigkeit von der Höhe zu verändern. Dies geschah exemplarisch erst durch einfache mathematische Operationen, die dann im Verlauf der Stunde durch komplexere ersetzt wurden. Der Vorteil, den Körper aus einzelnen Schichten aufzubauen, liegt darin, dass der 3D-Drucker seine Objekte auch durch Hinzufügen einzelner Schichten aufbaut. Die Schülerinnen und Schüler konnten so im Vorfeld sehen, wie aus einzelnen Schichten der Gesamtkörper entsteht.

2.2 Visuelle Programmierung mit Beetle Blocks

Mit der Einführung von Scratch [MRR10] wurde die Einstiegshürde für Programmieranfänger deutlich gesenkt, was insbesondere den Schülerinnen und Schülern an der Realschule sehr entgegen kommt. Durch die visuelle Darstellung der programmierten Objekte auf der Bühne wird die Programmierung im Sinne Paperts' Konstruktivismus unmittelbar erfahrbar, die multimediale Bandbreite ermöglicht den Schülerinnen und Schülern persönlich bedeutsame Ideen umzusetzen und die Folgen ihrer Programmierentscheidungen unmittelbar nachzuvollziehen. Durch die Repräsentation der Anweisungen als Blöcke werden zudem Syntaxfehler vermieden und die zur Verfügung stehenden Programmierkonstrukte sind unmittelbar zur Auswahl präsent. So wie Strecker [St15] auf der INFOS 2015 über den Einsatz grafischer Programmiersprachen im Abitur berichtete, überzeugt der visuelle Zugang aufgrund seiner Intuitivität und leichten Bedienung auch in der Realschule. Koschitz und Rosenbaum [KR12] übertragen nun die Konzepte und Bedienung von Scratch und die Idee der Turtle-Grafik unter Verwendung von Snap [HM10] in den dreidimensionalen Raum. Mit Beetle Blocks⁴ können mit Hilfe der algorithmischen Grundstrukturen dreidimensionale Objekte programmiert werden, die anschließend als 3D-Modell exportierbar sind und mittels eines 3D-Druckers ausgedruckt werden können. Algorithmische, mathematische und künstlerische Objekte können somit unmittelbar „erfassbar“ werden. Im Unterricht haben wir gute Erfahrungen gemacht mit Schmuck, Alltagsgegenständen wie Vasen und Schalen und künstlerischen Figuren (vgl. Abb. 7). Das Potenzial eines an solchen Zielen orientierten Unterrichts liegt darin, dass auch vordergründig nicht an der Programmierung interessierte Schülerinnen und Schüler durch die kreative Erstellung von Produkten, die so erstmal nicht im Informatikunterricht erwartet wird, motiviert werden können.

⁴ <http://beetleblocks.com/>

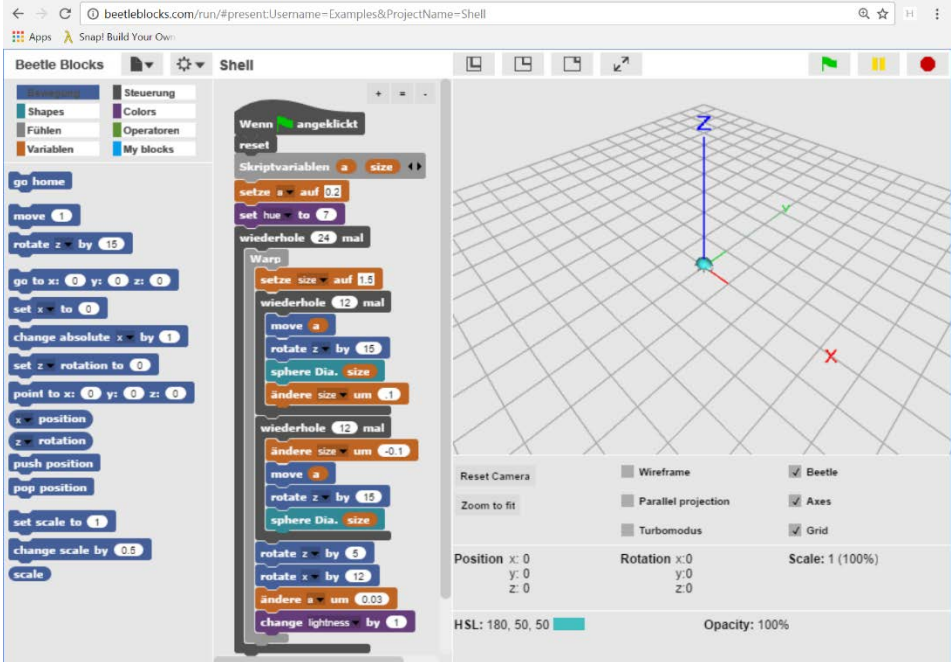


Abb. 2: Beetle Blocks.

3 Umsetzung

3.1 Kontext und Einordnung

Spätestens in der zehnten Klasse an bayerischen Realschulen lernen die Schüler im Modul „Modellierung und Codierung von Algorithmen“ des Lehrplans für das Fach Informatik die Grundstrukturen Sequenz, Auswahl und Wiederholung kennen und anzuwenden. Ein weiterer Punkt ist die Implementierung von Algorithmen mit einem geeigneten Programmierwerkzeug. Die Motivierung der Schüler ist hierbei ein wesentliches Anliegen des Unterrichts. Entsprechend haben sich die zum Erlernen verwendeten Programmiersprachen im Laufe der Zeit gewandelt. So wurde an unserer Schule zuerst die Programmiersprache Pascal verwendet. Um einen größeren Anreiz zu schaffen, sind wir dann auf Delphi umgestiegen. Mit Delphi haben die Schüler am Ende der Unterrichtssequenz einen Taschenrechner oder Getränkeautomaten programmiert und dabei die neu erworbenen Fähigkeiten in einem komplexeren Kontext angewandt. So ein selbst programmierter Taschenrechner motivierte schon etwas, weckte aber in der Schülergruppe auch keine Begeisterungstürme. Grund dafür war zum einen die doch sehr statische Interaktion mit selbigem und zum anderen die hohe Fehlerquote wegen syntaktischer Fehler. Geändert

hat sich das Ganze mit der Einführung der visuellen Programmiersprache Scratch. Mit dieser Programmiersprache konnten die Schüler sehr schnell und ohne syntaktische Fehler zu einem ansprechenden Ergebnis kommen. Die ersten kleinen Spiele waren sehr schnell programmiert und konnten mit nach Hause genommen werden. Mit Beetle Blocks verhielt es sich ebenso wie mit Scratch. Mit ein paar Befehlen konnten die Schülerinnen und Schüler sehr schnell den Käfer über den Bildschirm bewegen. Dies sorgte für Erheiterung in der Gruppe.

3.2 Unterrichtsziele und -gliederung

Ziel des Unterrichts war es, aus einfachen mathematischen Grundformen, wie z. B. Kreis, Quadrat oder gleichseitigem Dreieck, dreidimensionale Objekte zu entwickeln und diese zu programmieren. Als Vorlage konnten die Schüler sich an bekannten Gegenständen des täglichen Lebens orientieren, gleichfalls durften sie ihre 3D-Körper frei wählen, da sie unterschiedliche Vorstellungen hatten, welchen Gegenstand sie gern kreieren wollten. Ein Ziel hierbei war auch, dass den Schülerinnen und Schülern die Möglichkeit gegeben werde sollte, selbständig ihre eigene Kreativität zu nutzen und eigene Vorstellungen umzusetzen. Hierzu mussten die Schüler ihre Kenntnisse aus der Geometrie heranziehen. Für die Grundformen kamen neben dem allseits bekannten Satz des Pythagoras auch weitere grundlegende Kenntnisse, wie z. B. das Teilungsverhältnis der Höhen in einem gleichseitigen Dreieck, zum Einsatz. Mit diesem Wissen waren dann die Grundformen sehr schnell erstellt.

Für die dritte Dimension kamen mathematische Funktionen zum Einsatz: Mit Hilfe von Sinus, Kosinus und anderen Funktionen wurde die Größe der Grundformen in Abhängigkeit von der Höhe verändert, um den Objekten eine schöne geschwungene Form zu geben. In Abb. 3 sieht man eine eingefärbte Schnecke, die aus gleichseitigen Dreiecken aufgebaut ist. Bei der Schnecke wurde die Seitenlänge sukzessive konstant verkleinert.

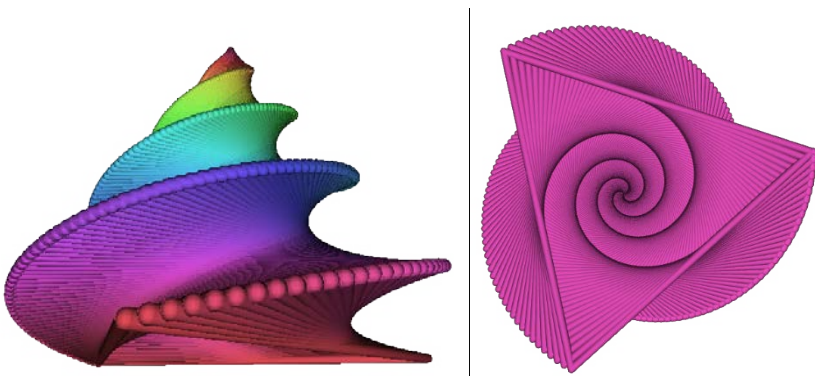


Abb. 3: Schnecke aus gleichseitigen Dreiecken.

Der Unterricht gliedert sich in 3 Phasen:

1. Kennenlernen der algorithmischen Grundstrukturen Sequenz und Wiederholung und Anwendung/Übung mit der Programmiersprache Beetle Blocks. Hierbei erstellen die Schülerinnen und Schüler einfache geometrische Figuren, wie z. B. Rechtecke oder Quadrate (3 Doppelstunden)
2. Erweiterung in die dritte Dimension, indem die Schüler die einfachen geometrischen Figuren als Turm aufbauen (1 Doppelstunde)
3. Mathematische Veränderung des Turms (3 Doppelstunden)

3.3 Unterrichtsverlauf und Beobachtungen

Bei der 10. Klasse handelte es sich um Schülerinnen und Schüler aus dem wirtschaftlichen Realschulzweig. Daher war die anfängliche Begeisterung der Schülerinnen und Schüler Programmieren zu lernen nicht gerade sehr groß. Dass das Ganze dann auch noch mit dem allseits geliebten Fach Mathematik verknüpft werden sollte, trug auch erstmal nicht gerade zur Motivationssteigerung bei. Dies zeigte sich z. B. in der Aussage der Schülerin C.: „Ich kann eh kein Mathe und jetzt muss ich auch noch damit Programmieren lernen.“ Entsprechend gestalteten sich die Anfangsstunden etwas zäh. Die Schülerinnen und Schüler sollten zuerst einmal den Begriff „Sequenz“ kennen und anwenden lernen. Dazu programmierten wir einfache geometrische Figuren, wie z. B. ein Rechteck. Etwas aufwändiger war die Programmierung eines gleichseitigen Dreiecks, da hier mit Hilfe des Satzes von Pythagoras die Koordinaten des dritten Eckpunkts erst einmal berechnet werden mussten.

Die vierte Doppelstunde war für die Schülerinnen und Schüler etwas leichter, da hier nur die bis dahin nicht verwendete Variable z neu hinzukam. Bis zu diesem Zeitpunkt sind sie noch ohne die Wiederholung ausgekommen. Bei der Erstellung des Turms haben die Schülerinnen und Schüler die Höhe ihres Turms durch wiederholte Erhöhung der Variable z mit einer Zählerschleife realisiert.

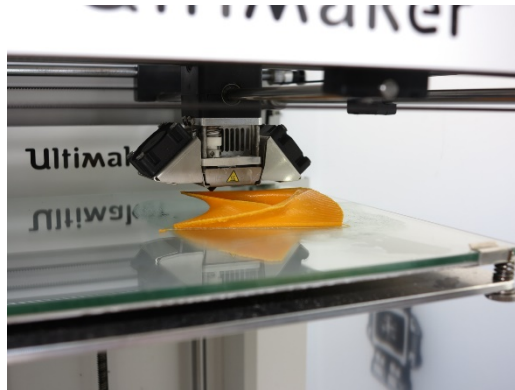


Abb. 4: Ausdruck eines Beetle Block-3D-Modells mit einem Ultimaker 2.

Ab der fünften Doppelstunde stieg die Motivation der Schülerinnen und Schüler deutlich an. Ich habe einen von mir mit dem 3D-Drucker ausgedruckten Körper mitgebracht und den Schülerinnen und Schülern versprochen, ihre Objekte auch mit dem 3D-Drucker auszudrucken. Nach dem konstruierenden Einstieg im Sinne des bottom-up verfolgte der weitere Unterricht nun einen experimentell-dekonstruierenden Ansatz. Hierzu wurden den Schülerinnen und Schülern verschiedene Methoden in Form von Beetle Blocks-Blöcken zum Experimentieren zur Verfügung gestellt, die zusätzlich zu den von den Schülerinnen und Schülern entwickelten Blöcken die Möglichkeit bereitstellten, dass die geometrischen Figuren noch um einen Winkel α gedreht werden konnten (vgl. Abb. 5).

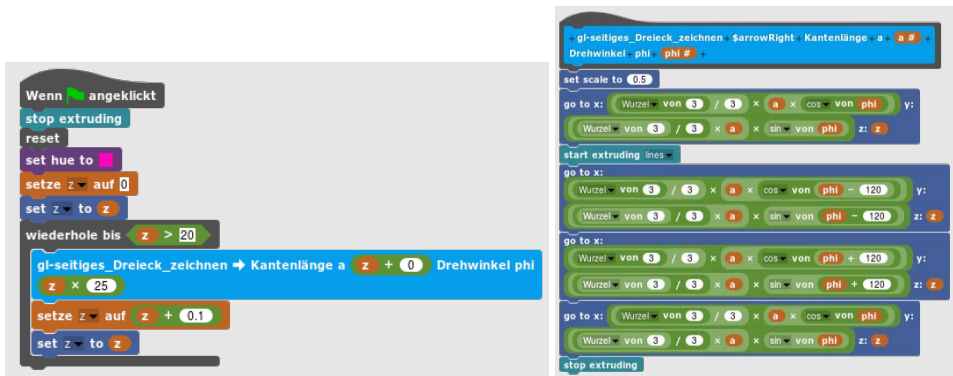


Abb. 5: Programmierung des Turms unter Verwendung eines vorgegeben Blocks (rechts).

Der Grund für die Bereitstellung war, dass die Schülerinnen und Schüler die zugrunde liegende mathematische Modellierung noch nicht im Mathematikunterricht behandelt hatten. Außerdem ist die Zeit mit sieben Doppelstunden knapp bemessen und es sollte den Schülerinnen und Schülern die Möglichkeit gegeben werden, mehr Zeit auf die Entwicklung ihrer eigenen 3D-Objekte zu verwenden. Dabei vertieften sie die bis dahin erlernte Vorgehensweise über den Aufbau der einzelnen Körper aus Schichten. Im Laufe der Zeit

änderte sich auch die Einstellung der Schülerin C. Mit Aussagen wie „Welche mathematische Funktion gibt es noch, die ich ausprobieren kann?“ wollte C. immer weitere Variationen ihres Turms ausprobieren. So kamen nicht nur einfach gedrehte Türme heraus, sondern auch z. B. kleine Schalen. Andere Schülerinnen und Schüler bauten ineinander geschachtelte Körper.

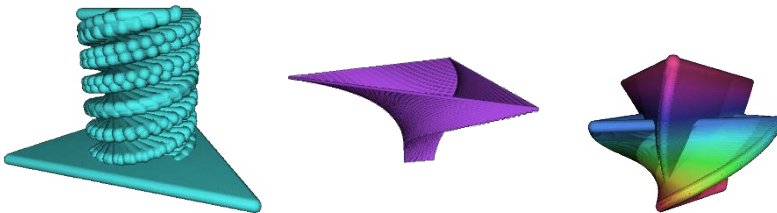


Abb. 6: Schülerergebnisse: Variationen der Türme.

Das in meinen Augen schönste Modell kreierte Schülerin C. Ein Schneckenhaus, welches aus gedrehten, gleichseitigen Dreiecken besteht, die nach oben immer eine kleinere Seitenlänge besitzen (vgl. Abb. 3). Alle Ergebnisse wurden dann mit einem 3D-Drucker ausgedruckt und konnten von den Schülerinnen und Schülern mit nach Hause genommen werden.

4 Fazit und Ausblick

Als Resümee lässt sich feststellen, dass die Motivation der Schülerinnen und Schüler stieg, als klar wurde, dass die 3D-Objekte, die sie programmierten, dreidimensional ausgedruckt werden sollten. Ihr Engagement, sich mit dem Erlernen von Programmierstrukturen zu beschäftigen, war vor diesem Zeitpunkt vergleichbar mit dem einer Gruppe, die mit Scratch das Programmieren erlernt. Auch in der Geschwindigkeit des Erlernens konnte kein Unterschied zu Scratch festgestellt werden.

Bei den ersten geometrischen Grundfiguren wurde es schwieriger. Die Schülerinnen und Schüler konnten zwar noch ohne Probleme die Koordinaten der Ecken eines Quadrates berechnen, aber bei einem gleichseitigen Dreieck ergaben sich gravierende Probleme. Hier muss für die Zukunft eine andere Vorgehensweise, mit einfacheren mathematischen Mitteln gefunden und erprobt werden. Die Anwendung der Mathematik wich hier doch von dem gewohnten Umgang im Mathematikunterricht ab. Vielleicht haben an dieser Stelle Schülerinnen und Schüler aus dem mathematischen Zweig oder am Gymnasium weniger Schwierigkeiten. Eine Gruppe aus zwei Schülern ist hier etwas anders vorgegangen. Anstatt die Koordinaten zu berechnen, haben sie ein gleichseitiges Dreieck auf einem Blatt Papier konstruiert, ungefähr in den Schwerpunkt des Dreiecks den Ursprung des Koordinatensystems gelegt und die Koordinaten des gleichseitigen Dreiecks abgelesen. In der

ritten Dimension konnten die beiden Schüler dann aber nicht mehr die Größe der Seite verändern.



Abb. 7: Vase, Kunstobjekt und Schnecke aus dem 3D-Drucker.

Der Weg in die dritte Dimension war hingegen wieder einfacher. Da die Grundfiguren so programmiert waren, dass ihre Größe von der Seitenlänge abhängt, konnte diese sehr leicht mit Hilfe von mathematischen Funktionen verändert werden. Die Schülerinnen und Schüler ließen hier ihrer Kreativität freien Lauf. Sie konnten dabei die Funktionen, die sie bis jetzt kennengelernt hatten, anwenden. In diesem Schritt variierten ein paar Schüler die Dicke der Linien, die vom Käfer gezeichnet wurden. Eine andere Schülergruppe veränderte mit der Höhe auch die Farbe der gezeichneten Linien. Zwei Schülerinnen haben ihre 3D-Körper mit einer Bodenplatte versehen, um eine Vase zu erhalten. Dieser Teil hat den Schülerinnen und Schülern besonderen Spaß gemacht. Eine Verknüpfung mit anderen Modulen aus dem Lehrplan, z. B. mit Arduinos (Modul: Mikrocontroller) als Steuerung für eine Leuchte (vgl. Abb. 7), wäre gut vorstellbar.

Für die Wiederholung der Unterrichtssequenz empfiehlt es sich, die Erstellung der Grundfiguren zunächst mit einfacheren mathematischen Mitteln auszuprobieren. Die Verknüpfung der Höhe mit mathematischen Funktionen hat sich bewährt. Hier sehen die Schülerinnen und Schüler, dass die Mathematik nicht nur trocken ist, sondern auch eine Anwendung hat, um der eigenen Kreativität Raum zu geben und neue Gegenstände zu entwickeln. Ein weiterer neuer Ansatzpunkt wäre, reale Gegenstände (vgl. Salzstreuer) in mathematische Grundkörper zu zerlegen und diese nachzuprogrammieren. In diesem Ansatz wäre wieder eine Verknüpfung zum Modul „Technisch Zeichnen“ gegeben.

Literaturverzeichnis

- [HM19] Harvey, B. and Mönig, J.: Bringing “no ceiling” to scratch: Can one language serve kids and computer scientists. Proc. of Constructionism Paris, 2010.
- [KR12] Koschitz, D., and Rosenbaum, E.: Exploring algorithmic geometry with “beetle blocks:” a graphical programming language for generating 3d forms. 15th International Conference on Geometry and Graphics Proceedings, Montreal. 2012.
- [MRR10] Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E.: The scratch programming language and environment. ACM Transactions on Computing Education (TOCE), 10(4), 16, 2010.
- [Pa82] Papert, S.: Mindstorms: Kinder, Computer und Neues Lernen. Birkhäuser Verlag, Basel, 1982.
- [St15] Strecker, K.: Grafische Programmiersprachen im Abitur. In: Gallenbacher, J. (Hrsg.): INFOS 2015: Informatik allgemeinbildend begreifen (16. GI-Fachtagung Informatik und Schule, Darmstadt). Bonn: Köllen, 2015.

Nutzung eines Robotiksystems zur Förderung der Berufssprache Deutsch

Michael Brinkmeier¹, Stanislav Pisarenko²

Abstract: Schulen und Lehrkräfte stehen vor der Aufgabe, jugendlichen Flüchtlingen sowohl sprachliche als auch berufliche Kompetenzen zu vermitteln. Im Projekt *Berufssprache Deutsch* wurden Methoden und Konzepte entwickelt, die beides miteinander verknüpfen. Darauf basierend entstand eine Unterrichtsreihe, die informatische Inhalte mit Sprachunterricht verbindet, und in einer Lerngruppe praktisch erprobt wurde. Die Verwendung des Roboters Thymio II und einer visuellen Programmiersprache ermöglichte dabei die explorative Vermittlung von Prinzipien der Programmierung und Automaten. Durch die gezielte Koppelung von fachlichen Aufgaben mit sprachlichen Aufgaben und Plenumsdiskussionen zu den erarbeiteten Lösungen konnten die Schülerinnen und Schüler ihre fachlichen und sprachlichen Kompetenzen koppeln und als Experten miteinander kommunizieren. Problem- und handlungsorientierte Aufgaben förderten zudem die Kommunikation zwischen den Teilnehmern.

Keywords: Roboter, Thymio, Robotik im Unterricht, Sprachlernklassen, Mealy-Automaten, Visuelle Programmiersprachen, Berufssprache Deutsch

1 Einführung

Die erhöhte Anzahl von jugendlichen Flüchtlingen, die in den letzten Jahren nach Deutschland kam, stellt das Bildungssystem und konkret die Schulen und Lehrkräfte vor große Herausforderungen. Einerseits müssen die Schülerinnen und Schüler die deutsche Sprache erlernen. Zum anderen sollen sie möglichst schnell fachlich und beruflich qualifiziert werden. Letzteres ist aber aufgrund der geringen Sprachkenntnisse und der gebrochenen Bildungsbiographien besonders schwierig. Integrative Konzepte, wie das des Projektes *Berufssprache Deutsch* [Be13], bieten hier Ansätze, um parallel fachliche und sprachliche Kompetenzen zu vermitteln.

Die hier beschriebene Einheit verwendet diesen Ansatz, um in einer Sprachlernklasse³ einerseits Inhalte der Informatik und Technik zu vermitteln und gleichzeitig den fachlichen Bezug für Sprachübungen zu nutzen. Ein besonderer Schwerpunkt soll dabei auf der Erzeugung von fachbezogenen Anlässen zur verbalen Kommunikation liegen, indem die

¹ Universität Osnabrück, Institut für Informatik, Wachsbleiche 27, 49040 Osnabrück, mbrinkmeier@uni-osnabrueck.de

² BBS Technik Cloppenburg, Lankumer Feldweg, 49661 Cloppenburg, Stanislav.Pisarenko@bbst-clp.de

³ Details dazu in Abschnitt 4.

Schülerinnen und Schüler gemeinsam an Problemlösungen arbeiten. Dabei ist es von großer Bedeutung, dass sie sich die fachlichen Aspekte ohne große sprachliche Barrieren erschließen können. Um dies zu erreichen, wird der Roboter Thymio II [th17b] verwendet. Er kann mittels der rein ikonographischen und visuellen Programmiersprache VPL programmiert werden, wodurch ein sehr leichter Einstieg ermöglicht wird. Dadurch können die Schülerinnen und Schüler sich im ersten Schritt auf die Aneignung der fachlichen Kompetenzen konzentrieren. Im Anschluss erfolgt jeweils eine sprachliche Fixierung und Vertiefung der gelernten Inhalte, entweder in Form kurzer Texte, einer verbalen Diskussion oder Übungen zu grammatikalischen Strukturen, wie z. B. Konditionalgefügen. Durch die Bearbeitung offener und experimenteller Aufgabestellungen ergeben sich für die Schülerinnen und Schüler Gelegenheiten zur fachbezogenen, verbalen Kommunikation. Daher ist ein Schwerpunkt der Evaluation die Beobachtung, ob diese tatsächlich genutzt werden, und ob sie zu einer aktiven Verwendung von Fachbegriffen beiträgt.

Die Einheit wurde in drei Tagesblöcken von jeweils sechs Stunden an einer berufsbildenden Schule durchgeführt. Zur Evaluation wurden verschiedene qualitative Methoden herangezogen. Neben Beobachtungsbögen zur Häufigkeit der verbalen Kommunikation, die von zwei Lehrkräften ausgefüllt wurden, wurde an jedem Tag das Feedback der Schülerinnen und Schüler sowie eine abschließende Beurteilung eingesammelt. Ergänzt wurde die Evaluation durch ein Kurzinterview mit einer über die gesamte Zeit anwesenden, beobachtenden Lehrkraft.

2 Berufssprache Deutsch

Das Projekt *Berufssprache Deutsch* des Staatsinstituts für Schulqualität und Bildungsforschung München (ISB) beschäftigt sich mit der berufsspezifischen Sprachförderung von Jugendlichen (16-20 Jahre) in der Ausbildung oder in Klassen der Berufsorientierung und Berufsvorbereitung [Be13, SRG13]. Im Rahmen des Projektes wurde eine Reihe von Materialien und Konzepten entwickelt, die die Grundlage der hier dargestellten Unterrichtssequenz bilden [St15, Le10]. Ihre vier Hauptkomponenten sind in Tab. 1 dargestellt (vgl. [Be13, S.2]).

Das **Sprachhandeln** basiert auf einer beruflichen oder fachlichen Handlungssituation in Kombination mit sprachlichen Anforderungen. Die Lerngruppe soll sich durch die motivierend wirkende, beruflich orientierte Aufgabenstellung sprachlich ausdrücken. Über die **Unterrichtsgestaltung** unterstützt die Lehrkraft die Lernenden mit einer bedarfsgerechten, kontinuierlichen und zielgerichteten Methodik. Die Formulierung von Aufgaben, Rückmeldungen und Lösungen muss in einer **sprachsensiblen Unterrichtssprache** erfolgen. Dies wird unter anderem durch einheitliche Arbeitsaufträge, Verwendung kurzer Sätze und alltagssprachlicher Wendungen erreicht [St15, S. 10]. Die Identifikation von **Förderschwerpunkten** ermöglicht an die Lerngruppe angepasste Hilfestellungen. Im Fokus stehen hierbei die Art der Aufgabenstellung, das Lerntempo und Formen des kooperativen Lernens.

Integrierte Sprachbildung	
Sprachhandeln: Handlungs- und Realitätsbezug (sprachliches Wissen und kommunikatives Handeln)	Sprachsensible Unterrichtsgestaltung durch angepasste Methodik (Methoden, Strategien, Arbeitstechniken)
Konzentration auf Förderschwerpunkte (Umgang mit Heterogenität, Binnendifferenzierung)	Sprachsensible Unterrichtssprache (Aufgabenstellung, Rückmeldung)

Tab. 1: Unterrichtskonzept Berufssprache Deutsch

In [St15] werden zusätzliche Aspekte thematisiert, die im Rahmen der vorliegenden Arbeit von Bedeutung sind. Streinz weist darauf hin, dass in Fällen von Flüchtlingen Traumata vorliegen können, die ein konzentriertes Lernen erschweren [St15, S. 8]. Spielerische Herangehensweisen können dies durchbrechen und die Kreativität der Lerner gezielt fördern. Hinzu kommt, dass ein nicht zu vernachlässigender Anteil der Jugendlichen bislang noch nie beschult worden ist oder nur Frontalunterricht kennengelernt hat. Dadurch ergeben sich besondere Probleme bei der selbstständigen Arbeit und offeneren Aufgaben.

In [Le10] wird eine Reihe von Methoden zur Kombination von Fach- und Sprachlernen vorgestellt. Leisen formuliert dort auch drei Grundthesen für sprachsensiblen Fachunterricht [Le10, S. 6, 22]:

- Die Lerner werden in fachlich authentische, bewältigbare Sprachsituationen gebracht.
- Die Sprachanforderungen liegen knapp über dem individuellen Sprachvermögen.
- Die Lerner erhalten so wenige Sprachhilfen wie möglich, aber so viele, wie individuell zum erfolgreichen Bewältigen der Sprachsituationen nötig.

Im informatischen Bereich ist außerdem der Wechsel zwischen sprachlichen, symbolischen und mathematischen Darstellungen von fachlichen Sachverhalten von Bedeutung:

Die Methode Wechsel der Darstellungsform⁴, hilft den Anschauungsgrad über verbalsprachliche Texte zu erhöhen und die Nutzung wird aus fachlichen, didaktischen, methodischen, lernpsychologischen und pädagogischen Gründen zwingend empfohlen (vgl. [Le10, S.37]).

3 Der Thymio

Die Behandlung informatischer Themen ist in der Regel sehr sprachlastig. Dies beruht zum einen auf der Verwendung von textuellen Programmiersprachen, als auch der notwendigen Beschreibung von algorithmischen Abläufen oder Funktionsweisen informatischer Systeme. Daher erscheint eine möglichst sprachunabhängige Einführung schwierig. Dies ist auch für die meisten grafischen Programmiersprachen der Fall.

⁴ Damit ist der Übergang zwischen der sprachlichen, symbolischen und mathematischen Darstellung gemeint.



Abb. 1: Der Thymio II und eine Regel in VPL: **Wenn** „links“gedrückt wird, **dann** leuchte rot.

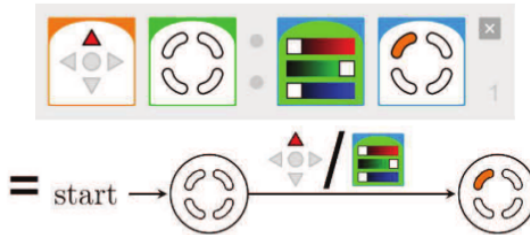


Abb. 2: Umsetzung einer VPL-Regel in den Übergang eines Mealy-Automaten

Im Rahmen der hier vorgestellten Einheit wird der Thymio II [th17b] – ein für didaktische Zwecke entworfener Roboter – verwendet, der bereits erfolgreich eingesetzt wurde [Ma15, Ma14, Ro15]. Insbesondere die für ihn entworfene grafische Programmiersprache VPL [th17a] zeichnet sich dadurch aus, dass sie vollständig auf Schriftsprache verzichtet. Die Programmierung des Roboters erfolgt in der einfachsten Variante in Form von Konditionalgefügen, die aus einer Bedingung und einer sich daraus ergebenden Aktion bestehen (vgl. Abb. 1). Auch wenn diese Form der Programmierung als sehr einfach erscheint, bietet sie bereits eine Vielzahl von Möglichkeiten. Durch die Verwendung eines internen Speichers und LEDs zur Visualisierung werden die Möglichkeiten deutlich erweitert und es wird möglich, das Verhalten des Thymio durch Mealy-Automaten zu modellieren [Wi16] (vgl. Abb 2).

4 Lerngruppe

Die Lerngruppe bestand im Untersuchungszeitraum aus neun jugendlichen Flüchtlingen, vier Mädchen und fünf Jungen, aus verschiedenen Ländern. Vier von ihnen waren seit ungefähr einem halben Jahr in der Gruppe, vier seit ca. zwei Monaten und einer erst seit wenigen Wochen. Die schulische Vorbildung war sehr inhomogen, ebenso wie die Altersstruktur (16-20 Jahre). Der Unterricht fand in wöchentlichen Blöcken von je sechs Schulstunden statt. Ziel war es, den Schülerinnen und Schülern praktische Erfahrungen im Bereich der technischen Berufe und die Berufs- und Fachsprache Deutsch zu vermitteln. Die Lerngruppe wurde regulär von zwei Lehrkräften unterrichtet. Die beschriebene Sequenz wurde von einer der Lehrkräfte (dem zweiten Autor) in einem Zeitraum von drei aufeinander folgenden Wochen durchgeführt, während die andere Lehrkraft nur als Beobachter fungierte.

Alle Schülerinnen und Schüler waren nicht in der Lage, flüssig auf Deutsch zu kommunizieren oder frei zu sprechen. Sie konnten einfache alltägliche Ausdrücke und Sätze verstehen und formulieren. Fach- und Berufssprache war ihnen weitgehend unbekannt. Das Erkennen und Verwenden von grammatikalischen Strukturen fiel ihnen sehr schwer. Außer grundlegenden Anwenderkenntnissen hatten alle Schülerinnen und Schüler keinerlei Erfahrung mit Informatiksystemen. Keiner von ihnen hatte vorher programmiert oder mit Robotern gearbeitet.

5 Methodik der Evaluation

Aufgrund der geringen Zahl der Teilnehmer wurden bei der Evaluation der Intervention qualitative Methoden verwendet. Im Rahmen einer teilnehmenden Beobachtung [DBP16, S. 323] durch die zwei Lehrkräfte wurde dabei ein besonderer Fokus auf die Häufigkeit und die Qualität der Kommunikation gelegt. Beide Beobachter protokollierten ihre Beobachtungen auf einem Erfassungsbogen. Dabei wurde aufgaben- bzw. phasenweise erfasst, ob eine fachbezogene Kommunikation stattfand, ob Fachbegriffe benutzt wurden oder ob fachfremde Themen erörtert wurden. Um eine Beeinflussung des Ergebnisses durch die Beobachtung auszuschließen, wurde den Schülerinnen und Schülern nicht mitgeteilt, wo der Schwerpunkt der Beobachtung lag [At08, S. 101].

Am Ende jedes der drei Tage war geplant, die Meinung der Schülerinnen und Schüler mittels Kartenfeedback abzufragen [Me15]. Dabei sollten sie kurze, schriftliche Antworten zu vorformulierten Fragen geben. Bei der ersten Durchführung wurde jedoch festgestellt, dass lediglich kurze, nicht wesentlich von vorgegebenen Phrasen abweichende Formulierungen, wie z.B. „weil ich gut finde“, genutzt wurden. Ausführlichere Antworten wurden nicht gegeben. Da die Verbalisierung Sprachanfängern deutlich leichter fällt als das Verschriftlichen [Le10, S. 22], wurden daher an den beiden folgenden Tagen die Fragen projiziert und die Schülerinnen und Schüler nacheinander befragt. Am dritten Tag wurde diese Befragung durch eine Gruppendiskussion über die gesamte Einheit ergänzt und die Äußerungen protokolliert.

Im Anschluss an die Einheit wurde ein Experteninterview mit der zweiten, beobachtenden Lehrkraft geführt [DBP16, S.365]. Da diese Lehrkraft – gemeinsam mit der durchführenden Lehrkraft – die Klasse regulär unterrichtet, konnte so eine unabhängige Beurteilung der erzielten Effekte erreicht werden. Es handelte sich um ein halb-strukturiertes Interview [DBP16, S. 372], bei dem zwei Fragen gestellt wurden:

- Wie ist der Gesamteindruck hinsichtlich der Motivation?
- Eignet sich der Thymio, um in einer sprachinhomogenen Schülergruppe eine anlass- und themenbezogene Kommunikation zu fördern?

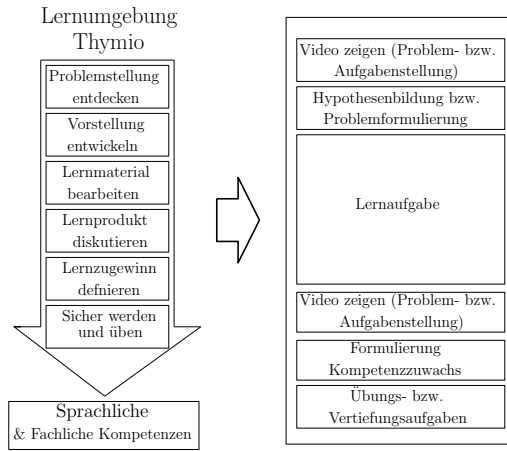


Abb. 3: Schrittfolge im Lernprozess basierend auf Leisen [Le10, S. 74]

6 Der Unterrichtsaufbau

Jede der drei evaluierten Sitzungen, im Umfang von jeweils sechs Schulstunden, wurde entlang der von Leisen (vgl. [Le10, S. 74]) entwickelten, lernpsychologisch begründeten Schrittfolge geplant und durchgeführt (Abb. 3).⁵ Damit wird beabsichtigt, den fachlichen und den sprachlichen Kompetenzzuwachs miteinander zu verbinden. In der ersten Phase sollen die Schülerinnen und Schüler die *Problemstellung entdecken*⁶. Dabei wird gezielt das nonverbale Medium des Videos genutzt. Es wird eine Problemsituation gezeigt, die von den Schülern gelöst werden. Das Video endet in allen Fällen mit einer kurzen, schriftlichen Frage, die als Impuls für die folgenden Arbeitsphasen dient. Im Anschluss daran wird der Arbeitsauftrag von den Schülern selbst schriftlich ausformuliert. Um dies zu unterstützen, erhielten die Schülerinnen und Schüler gestufte Hilfestellungen in Form von Wortlisten oder Blockdiagrammen (vgl. Abb. 4). Die in Einzelarbeit erarbeiteten Beschreibungen werden mit einem Partner abgeglichen und anschließend im Plenum vorgestellt.

Durch die Verschriftlichung wird die *Entwicklung von Vorstellungen* und eine erste Hypothesenbildung hinsichtlich der Problemlösung gefördert. Diese wird in den ersten Schritten der *Bearbeitung des Lernmaterials* fortgeführt und um praktische Tätigkeiten erweitert. Die Lernaufgaben sind dabei entdeckend, experimentell und handlungsorientiert gestaltet. Sie orientieren sich zum Teil an den in [Ro15] vorgestellten Aufgaben. Entsprechend der Struktur der grafischen Programmiersprache VPL werden dabei häufig Konditionalgefüge (Wenn-dann-Sätze) genutzt (vgl. Abb. 5). Im Anschluss an die Bearbeitung der Lernaufgabe werden die erarbeiteten Lösungen im Plenum *diskutiert* und mit dem ursprünglichen Video abgeglichen.

⁵ Die Materialien stehen unter <http://inf-didaktik.rz.uos.de/thymio/unterricht/sprachlernklassen.php> zur Verfügung.

⁶ Vgl. in Abb. 3 dargestellte Schrittfolge



grün. - Bei - Boden - leuchten - der - hellen - Roboter - leuchten - der - Bei - Boden - dunklen - Roboter - rot.

Abb. 4: Beispiel für Blockdiagramme und Wortgeländer

<u>Wenn</u> der Thymio ein Objekt vor sich entdeckt,	•	• <u>dann</u> dreht er sich nach links.
<u>Wenn</u> der Thymio ein Objekt seitlich rechts entdeckt,	•	• <u>dann</u> dreht er sich nach rechts.
<u>Wenn</u> der Thymio nichts erkennt,	•	• <u>dann</u> fährt er zurück.
<u>Wenn</u> der Thymio ein Objekt seitlich links entdeckt,	•	• <u>dann</u> fährt er nach vorne.

Abb. 5: Auszug aus dem Arbeitsmaterial des ersten Tages

Auf fachlicher Ebene wird das *Sicher werden und üben* durch sich anschließende, offenere Aufgaben realisiert. Diese greifen die vorher gelernten neuen Aspekte auf und nutzen Sie zur Lösung des eingangs vorgestellten oder eines neuen Problems. Die Kommunikation untereinander bildet den – für die Schülerinnen und Schüler nicht expliziten – Schwerpunkt. Der Aspekt *Sichern und Üben* ist daher nur zum Teil auf den fachlichen Anteil bezogen. Ergänzend soll insbesondere gesprochene Sprache als Werkzeug während des Problemlöseprozesses genutzt werden. Die fachlichen Inhalte bieten die Gelegenheit zum Experten zu werden und anschließend das eigene Handeln zu reflektieren und die eigenen Lernzugewinne und Problemlösungen zu formulieren. Dadurch sollen gezielt sprachliche Einzelkompetenzen, wie z.B. Texte planen, umstrukturieren und schreiben, sowie Lesekompetenz gefördert werden.

7 Die Ergebnisse der Evaluation

Eines der Hauptziele der Einheit war es, die Schülerinnen und Schüler dazu zu motivieren, über fachliche Inhalte zu kommunizieren. Die von der durchführenden und der beobachteten Lehrkraft ausgefüllten Beobachtungsbögen⁷ zeigen, dass dieses Ziel erreicht wurde. Dabei variierte die Anzahl der Gruppen, in denen über die Inhalte gesprochen wurde, mit der Art der Aufgabe. Bei solchen, die eine Verschriftlichung erforderten, war der Effekt schwächer, während insbesondere die experimentellen, explorativen und offeneren Aufgaben die Kommunikation deutlich förderten. Es wird auch deutlich, dass die Nutzung von Fachbegriffen anfänglich Schwierigkeiten bereitete. So wurden sie am ersten Tag innerhalb keiner Gruppe verwendet, am zweiten und dritten Tag aber durchgängig in ein bis drei Gruppen. Die Beobachter stellten auch fest, dass Gespräche über inhaltsfremde Themen nur stattfanden, wenn eine Gruppe die Arbeit bereits beendet hatte und auf die anderen wartete. Dies trat nur am dritten Tag ein.

⁷ Die Beobachtungsbögen stehen unter <http://inf-didaktik.rz.uos.de/thymio/unterricht/sprachlernklassen.php> zur Verfügung.

Tag	Kurzcharakteristik	Aufgabe(n)
1. Tag	Erkunden der Verhaltensweisen des Roboters, Konditionalgefüge	Aufgaben 1-4
	Einführung in die Programmierung, Ereignisse	Aufgaben 5 & 6
	Erstellen eines textuell beschriebenen Programms	Aufgabe 7
2. Tag	Entfernungs- und Bodensensoren	Aufgaben 1-4
	Implementieren im Video gezeigten und textuell beschriebenen Verhaltens (Bodensensoren)	Aufgabe 5
	Implementierung eines Linienfolgers	Aufgabe 6
3. Tag	Beschreibung von Vorgängen mit Zuständen (Schranke und Farbwechsel)	Aufgaben 1-2
	Programmierung einer Sequenz von Farben unter Verwendung von Zuständen und Darstellung als Übergangsgraph	Aufgabe 3, 4 & 7
	Implementieren des Programms für die Schranke	Aufgabe 8

Tab. 2: Der inhaltliche Verlauf der Intervention

Nachdem am ersten Tag festgestellt wurde, dass die Befragung in schriftlicher Form wenig ergiebig war, wurde auf eine mündliche Form gewechselt, bei der die Fragen der Gruppe vorgelesen und anschließend die Antworten abgefragt wurden. Die Verwendung von Videos zur Beschreibung der Aufgabenstellung wurde dabei einhellig als sehr positiv empfunden. Auch die explorativen Aufgaben, bei denen das Verhalten des Roboters durch Experimenten mit kleinen Programmen erkundet wurde, wurden als sehr positiv bewertet. Der dafür genannte Grund war der geringe Schwierigkeitsgrad der Aufgaben auf technischer Seite.

Aufgaben, die eine Verschriftlichung erforderten, wurden insgesamt als sinnvoll aber sehr schwer empfunden. Die Schülerinnen und Schüler nutzten dabei häufig die angebotenen Lernhilfen. Dies betraf insbesondere Aufgaben, bei denen nicht nur das beobachtbare Verhalten des Roboters beschrieben werden sollte, sondern auch inhaltliche Aspekte behandelt wurden. So fiel es den Schülerinnen und Schülern besonders schwer, Hypothesen über die Funktionsweise von Sensoren zu entwickeln. Dies kann zum einen auf die Schwierigkeit der Verschriftlichung, zum anderen auf fachliche Unsicherheit zurückgeführt werden. In allen Fällen konnte durch Gespräche im Plenum allerdings gemeinsam eine sinnvolle und korrekte Antwort formuliert werden. Dadurch war eine Konzentration auf die sprachlichen Aspekte möglich.

Offeneren Aufgaben (beschriebenes Verhalten implementieren, Linien folgen, Schranke implementieren) wurden insgesamt als sehr schwer empfunden. Der erste Fall, der Umsetzung eines Textes in ein Programm, fiel den Schülerinnen und Schülern anfänglich besonders schwer, da die Konditionalgefüge dort nicht in der bisher verwendeten Wenn-dann-Form verwendet wurden, sondern in Soll-wenn-Form (Der Thymio soll ..., wenn ...). Nachdem das Prinzip erläutert wurde, wurde die Aufgabe hingegen als sehr einfach betrachtet. Die entsprechende Aufgabe des zweiten Tages wurde ebenfalls wieder als sehr schwer empfunden, jedoch gelang es einer Gruppe, sie ohne Hilfestellung erfolgreich zu bearbeiten. Die Implementierung des Linienfolgers war den Schülerinnen und Schülern nur mit Hilfestellung möglich. Dennoch erschienen sie hochgradig motiviert und testeten häufig ihre Programme. Ähnlich verhielt es sich mit der Realisierung eines Schrankenroboters am dritten Tag. Die Umsetzung in eine Abfolge von Zuständen fiel den Schülerinnen und Schülern sehr schwer und wurde gemeinsam im Plenum entwickelt und anschließend umgesetzt.

In der abschließenden Gruppendiskussion äußerten sich die Schülerinnen und Schüler sehr positiv über die Beschreibungen der Verhaltensweisen des Thymios. Dabei empfanden sie es als besonders nachhaltig, erst selbstständig zu formulieren und dies anschließend im Plenum vorzustellen und an einer gemeinsamen, grammatikalisch und semantisch korrekten Verschriftlichung zu arbeiten. Die Frage ob der Thymio sich als Kommunikationsanlass und als Werkzeug zum Erlernen der deutschen Sprache eignen würde, äußerten sich alle acht anwesenden Teilnehmer der Gruppe positiv. Fünf empfanden die Verwendung eines solchen Werkzeugs als sehr motivierend und besser als den „üblichen“ Sprachunterricht. Sie führten dies insbesondere auf den explorativen und spielerischen Charakter des Thymios zurück, sowie auf die Motivation etwas über Programmierung lernen zu wollen. Die drei anderen Gruppenmitglieder empfanden die Arbeit mit dem Thymio als mindestens so gut wie einen abwechslungsreichen Grammatik- und Spielunterricht. Alle empfanden die Einheit aber als deutlich nachhaltiger als reinen Theorieunterricht.

Der insgesamt positive Eindruck wurde durch das Interview der beobachtenden Lehrkraft bestätigt. Die Verwendung des Thymios und von Videos führte ihrer Ansicht nach zu einer deutlich spürbaren Steigerung der intrinsischen Motivation. Diese konnte zwar nicht über die gesamte Zeit aufrechterhalten werden, das wäre aber angesichts der Länge eines Tages (sechs Schulstunden, unterbrochen durch die üblichen Pausen) nicht anders zu erwarten. Die beobachtende Lehrkraft bewertete die Eignung des Thymios, um Kommunikationsanlässe zu schaffen und Sprachübungen inhaltlich zu verankern, als durchweg positiv. Besonders erstaunt äußerte sie sich darüber, dass die Schülerinnen und Schüler ihren aktiven Wortschatz sehr schnell um Fachbegriffe erweiterten und sie sowohl in der verbalen und schriftlichen Kommunikation verwendeten. Zum Abschluss äußerte die beobachtenden Lehrkraft die Absicht, künftig auch weiterhin mit ähnlichen Methoden, anderen Robotiksystemen und Programmierung in der Lerngruppe arbeiten zu wollen.

8 Fazit

Der Ansatz, ein Robotiksystem wie den Thymio im Sprachunterricht zu verwenden, erscheint im Anschluss an die hier beschriebene Einheit als tragfähiges und vielversprechendes Konzept. Er ermöglicht die sinnvolle inhaltliche Einbettung von Sprachübungen in einen informatisch-technischen Kontext. Dabei wurde allerdings im Vorfeld die Schwierigkeit, relativ kurze, fachbezogene Texte zu verstehen, unterschätzt. Auch die Bearbeitung von offeneren Aufgaben erforderte in der Regel zusätzliche Unterstützung, die bislang über eine Gruppendiskussion oder verbal über die Lehrkraft erfolgte. Insbesondere die Plenumsrunden wurden von den Schülerinnen und Schülern, auch im Zusammenhang mit anderen Aufgaben, wie z.B. der Verschriftlichung von Beobachtungen, als besonders nachhaltig erachtet. Die beobachtende Lehrkraft hob insbesondere die häufige Verwendung von Fachbegriffen hervor. Insgesamt wurde die Einheit von allen Beteiligten als motivierend und zielführend beschrieben.

Durch die kleine Anzahl an Teilnehmern können die Ergebnisse der Evaluation nur eine grundsätzliche Tendenz beschreiben. Um belastbare Resultate zu erzielen, müssen weitere Versuche mit angepassten Materialien durchgeführt werden. Zudem fand keine objektive Erhebung der vermittelten informatischen und sprachlichen Kompetenzen statt.

Literaturverzeichnis

- [At08] Atteslander, Peter; Cromm, Jürgen; Grabow, Busso; Klein, Harald; Maurer, Andrea; Siegert, Gabriele: Methoden der empirischen Sozialforschung. Erich Schmidt Verlag, Berlin, 2008.
- [Be13] Berufssprache Deutsch. Online unter: <http://www.isb.bayern.de/schulartsspezifisches/materialien/berufssprache-deutsch/> (Zuletzt aufgerufen am 14.09.2016), 2013.
- [DBP16] Döring, Nicola; Bortz, Jürgen; Pöschl, Sandra: Forschungsmethoden und Evaluation in den Sozial- und Humanwissenschaften. Springer, Berlin & Heidelberg, 2016.
- [Le10] Leisen, J.: Handbuch Sprachförderung im Fach: Sprachsensibler Fachunterricht in der Praxis. Varus, 2010.
- [Ma14] Magnenat, Stéphane; Shin, Jiwon; Riedo, Fanny; Siegwart, Roland; Ben-Ari, Morderchai: Teaching a Core CS Concept Through Robotics. In: Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education. ITiCSE '14, ACM, New York, NY, USA, S. 315–320, 2014.
- [Ma15] Magnenat, S.; Ben-Ari, M.; Klinger, S.; Sumner, R. W.: Enhancing Robot Programming with Visual Feedback and Augmented Reality. In: Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education. ACM, New York, NY, USA, S. 153–158, 2015.
- [Me15] Meyer, H.: Praxisbuch: Leitfaden Unterrichtsvorbereitung. Cornelsen Scriptor, Berlin, 2015.
- [Ro15] Roy, Didier; Gerber, Gordana; Magnenat, Stéphane; Riedo, Fanny; Chevalier, Morgane; Oudeyer, Pierre-Yves; Mondada, Francesco: IniRobot : a pedagogical kit to initiate children to concepts of robotics and computer science. In: RIE 2015. 2015.
- [SRG13] Sogl, P.; Reichel, P.; Geiger, R.: „Berufssprache Deutsch“ - Ein Projekt zur berufsspezifischen Sprachförderung im Unterricht an der Berufsschule bzw. Berufsfachschule in Bayern. bwp@Spezial, 6, 2013. Online unter: http://www.bwpat.de/ht2013/ft18/sogl_etal_ft18-ht2013.pdf (Zuletzt aufgerufen am 28.05.2017).
- [St15] Streinz, A.: Berufsschulpflichtige Asylbewerber und Flüchtlinge. ISB - Staatsinstitut für Schulqualität und Bildungsforschung, 2015. Online unter: https://www.isb.bayern.de/berufliche-schulen/materialien/baf_beschulung/ (Zuletzt aufgerufen am 26.09.2016).
- [th17a] thymio.org: Anleitung zum visuellen Programmieren. 2017. Online unter: <https://www.thymio.org/de:thymiovgl> (Zuletzt aufgerufen am 28.5.2017).
- [th17b] thymio.org: Thymio & Aseba. 2017. Online unter: <https://www.thymio.org/de:thymio> (Zuletzt aufgerufen am 28.5.2017).
- [Wi16] Winterboer, S.; Zilke, R.: Zustandsbasierte Modellierung und Implementierung von Schwarmverhalten im Unterricht. Masterarbeit, 2016. Masterthesis, Universität Osnabrück.

Schülerinteresse an Informatik und Informatikunterricht

Torsten Brinda,¹ David Tobinski,² Stefan Schwinem³

Abstract: Bislang ist noch relativ wenig empirisch erforscht, was das Interesse von Schülerinnen und Schülern an Informatik und Informatikunterricht weckt bzw. beeinflusst. In Betracht kommen dazu bspw. didaktische Entscheidungen seitens der Lehrkraft zu informatischen Kontexten, betrachteten Phänomenen, Situationen, beteiligten informatischen Inhaltsfeldern, erforderlichen Schülertätigkeiten und vielem mehr. Der vorliegende Beitrag berichtet von einer explorativen empirischen Studie, welche das Ziel hatte, ein Instrument zur Erhebung des Interesses von Lernenden am Fach Informatik, an informatischen Alltagserscheinungen, Situationen und Kontexten sowie den vorgesehenen Unterrichtsinhalten in den Sekundarstufen zu entwickeln und zu erproben. Weiterhin sollten erste Hinweise auf Unterschiede zwischen verschiedenen Schülergruppen herausgearbeitet werden. Dazu wurden Lernende verschiedener Schulformen und Jahrgangsstufen mittels eines standardisierten Online-Fragebogens schriftlich befragt. Ein insgesamt stärker ausgeprägtes Interesse zeigte sich bei männlichen Schülern sowie Lernenden, die angaben, einen informatikbezogenen Berufswunsch zu verfolgen, dass Informatik ihr Lieblingsfach sei oder gute bis sehr gute Mathematik- oder Informatiknoten zu haben.

Keywords: Schülerinteresse; Informatikinteresse; Interessensforschung; Sachinteresse; Fachinteresse; Fragebogen; empirische Studie; explorative Studie

1 Einleitung

Allgemeinbildender Schulunterricht hat u. a. das Anliegen, Lernende jeweils für die Auseinandersetzung mit fachlichen Gegenständen zu motivieren, zu interessieren – idealerweise sogar zu begeistern und darüber ein längerfristiges Domäneninteresse zu entfalten und zu festigen. Krapp und Prenzel geben einen Überblick über das Forschungsfeld Interesse im Bereich der Pädagogischen Psychologie [KP11]. Interesse kann aus der Interaktion eines Individuums mit seiner Umwelt entstehen und wird durch Werte und Gefühle des Individuums beeinflusst. Unterschieden werden dabei individuelles Interesse, das in der Persönlichkeitsstruktur des Individuums verankert ist, sowie situationales, durch externe Faktoren generiertes Interesse. Aus schulischer Perspektive möchte man erreichen, dass Schülerinnen und Schüler sich für ein Unterrichtsfach, dessen Inhalte und das Geschehen in der jeweiligen Unterrichtsstunde interessieren. Da Interesse in der Informatik bislang kaum empirisch untersucht wurde, verfolgte die vorliegende Arbeit (s. a. [Sc16]) das Ziel, einen Beitrag zur Beschreibung und zur Erhebung von Informatikinteresse zu leisten.

¹ Univ. Duisburg-Essen, Didaktik der Informatik, Schützenbahn 70, 45127 Essen torsten.brinda@uni-due.de

² Univ. Duisburg-Essen, Fak. f. Bildungswissensch., Universitätsstr. 2, 45127 Essen david.tobinski@uni-due.de

³ Univ. Duisburg-Essen, ICB, Schützenbahn 70, 45127 Essen stefan.schwinem@stud.uni-due.de

2 Verwandte Arbeiten

Eine bedeutende Arbeit aus dem Bereich der Interessensforschung stammt aus dem Unterrichtsfach Physik. In der sogenannten *IPN-Interessensstudie Physik* wurden das Sach- und Fachinteresse an Physik in verschiedenen Jahrgangsstufen der Sek. I über einen längeren Zeitraum mittels eines umfangreichen Fragebogens erhoben und analysiert [HHL98]. Basis der Untersuchung bildete ein dreidimensionales Interessensmodell mit den Dimensionen *Fachinhalte*, *Kontexte* und *Tätigkeiten*. Dieses Modell wurde auch für die vorliegende Arbeit zugrunde gelegt. In der internationalen *Relevance of Science Education (ROSE)*-Studie [HB07] wurden Ranglisten der interessantesten und uninteressantesten Themen mit Bezug zu den Naturwissenschaften erstellt – aus Informatikperspektive bemerkenswert ist der Befund, dass das Thema „Wie Computer funktionieren“ das für Jungen drittinteressanteste Thema in dieser naturwissenschaftsbezogenen Untersuchung war. Für die Informatik liegt eine umfangreichere Studie zu „Schülerwünschen im Informatikunterricht“ vor (vgl. [Ba10]), die ebenfalls am IPN entstand. Die Wünsche von Schülerinnen und Schülern zum Unterricht wurden basierend auf dem zum Zeitpunkt der Studie gültigen Informatiklehrplan des Landes Schleswig-Holstein sowie den Empfehlungen zu Bildungsstandards Informatik für die Sek. I der Gesellschaft für Informatik e. V. (GI) [Ge08] erhoben. Weitergehende Untersuchungen zu einer topologischen Struktur von Informatikinteresse (wie in der IPN-Interessensstudie Physik) waren nicht enthalten.

3 Erhebung von Informatikinteresse

Obige Vorüberlegungen bildeten den Ausgangspunkt für die Entwicklung und Pilotierung eines Online-Fragebogens zur explorativen Erhebung von Informatikinteresse. Ein Online-Fragebogen wurde gewählt, um im Wesentlichen ortsunabhängig eine potenziell größere Gruppe von Teilnehmenden erreichen zu können. Der Aufbau des Fragebogens orientierte sich an [HHL98], Informatikinteresse wurde analog dazu in Fach- und Sachinteresse differenziert. Neben soziodemographischen Fragen und Fragen zur Belegung von Informatikunterricht war der Fragebogen strukturiert in Blöcke mit Aussagen zum Fachinteresse, zur etwaigen Wahlentscheidung für das Unterrichtsfach, zum Stellenwert von Informatik in der eigenen Freizeit, zum Interesse an lebensweltlichen Situationen und Kontexten mit Informatikbezug sowie zum Interesse an Gegenständen und Tätigkeiten innerhalb der Informatik. Diese waren jeweils jeweils mittels vierstufiger Ratingskala („trifft zu“ bis „trifft nicht zu“) hinsichtlich des damit verbundenen Interesses zu bewerten. Der Online-Fragebogen wurde mittels *LimeSurvey* erstellt (www.limesurvey.org). Zur Gewinnung von Teilnehmenden wurden über bestehende E-Mail-Verteiler für Informatiklehrkräfte in Nordrhein-Westfalen, Baden-Württemberg und Rheinland-Pfalz ein Anschreiben mit einer Erläuterung der geplanten Untersuchung, eine pdf-Fassung des Fragebogens sowie der Link zur Teilnahme verteilt. Weiterhin wurden regionale Schulen einbezogen, zu denen persönliche Kontakte bestanden. Die Erhebung fand dann in der zweiten Junihälfte 2016 statt. Insgesamt wurden dabei 172 Datensätze erfasst, von denen $N = 141$ auswertbar waren.

4 Ergebnisse einer ersten Pilotierung

Im Bereich *Fachinteresse an Informatik* wurden sechs von acht Items zwischen 56% und 73% mit „trifft zu“ oder „trifft eher zu“ beantwortet. Im Folgenden angegebene Prozentzahlen beziehen sich immer auf die Summe der Werte der beiden zustimmenden bzw. ablehnenden Kategorien. Zur *Fächerwahl* wurde Items (eher) zugestimmt, die Hilfe zur Selbsthilfe fokussierten (55%), das Verständnis informatischer Zusammenhänge adressierten (58%) oder den Erwerb von Softwarenutzungs Kompetenzen in den Blick nahmen (54%). Aussagen zur Auseinandersetzung mit Gefahren im Internet, Berufs- oder Studienentscheidungen, Fachbelegungen durch Geschwister, im Informatikbereich tätige Freunde oder Verwandte oder für andere Fächer erhoffte Vorteile wurden mehrheitlich (eher) abgelehnt oder ergaben ein uneinheitliches Bild. Eine Beschäftigung mit *Informatik im Alltag* war für die Teilnehmenden eher von untergeordneter Bedeutung – alle dsbzgl. Items wurden von 60% bis 86% der Teilnehmenden (eher) abgelehnt. Bzgl. der *Situationen und Kontexte* bewerteten Lernende in der *Sek. I* ($n=73$) die Auseinandersetzung mit Schadsoftware (im Mittel der Summe aller zugeordneten positiven Aussagen: 57%), Computerspielen (62%) sowie Smartphones und Tablets (53%) mehrheitlich als (eher) interessant, die entsprechende Beschäftigung mit Textverarbeitung (entsprechend negative Aussagen: 60%) sowie sozialen Netzwerken (54%) jedoch als (eher) von geringem Interesse. Das erfragte Vorwissen hatte auf diese Bewertungen keinen erkennbaren Einfluss: das Vorwissen zu Textverarbeitung, Computerspielen sowie Smartphones und Tablets wurde als eher hoch eingeschätzt, das zu den übrigen Themen als eher gering. Explizit gefragt nach verschiedenen Einflussfaktoren auf das jeweilige Interesse zeigte sich, dass im Zusammenhang mit Textverarbeitung die vermutete Relevanz für die eigene berufliche Zukunft (53%) und bei Computerspielen der Stellenwert im Freundeskreis (66%) sowie diesbezügliches Freizeitinteresse (59%) positiv auf das Interesse wirkten. Dass dsbzgl. Interesse durch die Thematisierung im Informatikunterricht oder Medienberichte induziert worden wäre, wurde mehrheitlich (eher) abgelehnt – das gilt auch analog für die *Sek. II*. Diese Teilnehmenden ($n=68$) bewerteten ihr Interesse an den Kontexten Computerspiele (im Mittel der Summe aller zugeordneten positiven Aussagen: 66%), Suchmaschinen (55%), Clouds (51%), Smartphones und Tablets (62%), Hausautomatisierung (48%, ca. 8% der Teilnehmenden gaben hierzu keine Antwort; bei den übrigen Kontexten ca. 1%-3%) insgesamt (eher) als groß, lediglich einige Items mit anwendungsbezogenen und gesellschaftlich-kulturellen Bezug (Kontexte Suchmaschinen, Clouds, Hausautomatisierung) zeigten ein unentschiedenes oder leicht negatives Bild. Gefragt nach verschiedenen *Inhalten und Tätigkeiten* im Informatikunterricht zeigte sich bei den Lernenden der *Sek. I* folgendes Bild: das Modellieren und Implementieren von Algorithmen und Programmen zur Lösung von Problemen – allein oder in Kooperation mit anderen –, das Verständnis der Funktionsweise von Informatiksystemen sowie die korrekte Verwendung informatischer Fachbegriffe wurden als (eher) interessant bewertet, Aussagen mit Bezug zum Inhaltsfeld Informatik, Mensch und Gesellschaft, z. B. zu sicheren Passwörtern, zum Verhalten in sozialen Netzwerken und zur Entwicklung der Informatik als (eher) wenig interessant eingestuft. Bei den Teilnehmenden in der *Sek. II* zeigte sich im Kern die gleiche Tendenz.

Lernende in der Sek. I zeigten sich insgesamt interessierter als Lernende in der Sek. II. Ein stärker ausgeprägtes Interesse hatten auch Lernende mit informatikbezogenem Berufswunsch, Lieblingsfach Informatik sowie (sehr) guten Mathematik- oder Informatiknoten. Bezüglich der Geschlechter war die Interessensbewertung häufig nicht signifikant unterschiedlich. Weiterhin zeigte sich bei den Teilnehmenden ein insgesamt eher geringes Interesse, sich in der Freizeit mit informatischen Themen zu befassen, es sei denn, hierbei war ein persönlicher Nutzen offensichtlich (Bedienung von Smartphones, Suchmaschinen).

5 Fazit und Ausblick

Wenn das eigene lebensweltliche Interesse im Vordergrund stand, beantwortete ein höherer Anteil Lernender die entsprechenden Items positiv. Gegenstände, die in der alltäglichen Lebenswelt der Lernenden keine besonders große Bedeutung besaßen, standen ebenfalls weniger in deren Interesse. Gesellschaftlich-kulturelle Aspekte trafen – im Gegensatz zu technischen und anwendungsbezogenen Aspekten – auf geringes Interesse. Mit der Studie liegt nun ein einmalig erprobtes Instrument vor, das im Weiteren bearbeitet und verfeinert werden muss. Die Auswahl der Situationen und Kontexte bedarf der weiteren Entwicklung, ebenso sollten zu bewertende Aussagen zu Unterrichtsinhalten noch systematischer variiert werden. Es wäre wünschenswert, zukünftig noch mehr über die Interessen von Schülerinnen und Schülern zu erfahren, um diese stärker bei didaktischen Entscheidungen berücksichtigen zu können. Das in dieser Untersuchung eher geringe Interesse an gesellschaftlich-kulturellen Aspekten sollte zur Prüfung und ggfs. Überarbeitung der zugeordneten Items führen, weiterhin zu der Erkenntnis, dass eine interessante Unterrichtsgestaltung hier besonders wichtig sein könnte.

Literaturverzeichnis

- [Ba10] Barthel, Hannelore: Informatikunterricht. Wünsche und Erwartungen von Schülerinnen und Schülern. Dissertation, Philosophische Fakultät, Christian-Albrechts-Universität, Kiel, Januar 2010.
- [Ge08] Gesellschaft für Informatik e.V.: Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I. LOG IN, 28(150/151):Beilage, 2008.
- [HB07] Holstermann, Nina; Bögeholz, Susanne: Interesse von Jungen und Mädchen an naturwissenschaftlichen Themen am Ende der Sekundarstufe I. Zeitschrift für Didaktik der Naturwissenschaften, 13:71–86, 2007.
- [HHL98] Hoffmann, Lore; Häusler, Peter; Lehrke, Manfred: Die IPN-Interessenstudie Physik. IPN, Kiel, 1998.
- [KP11] Krapp, Andreas; Prenzel, Manfred: Research on Interest in Science: Theories, methods, and findings. International Journal of Science Education, 33(1):27–50, January 2011.
- [Sc16] Schwinem, Stefan: Schülerinteresse im Informatikunterricht. Masterarbeit, Universität Duisburg-Essen, Essen, August 2016.

Workshops

Willkommen im Programmierzirkus – Ein Programmierkurs für Grundschulen

Katharina Geldreich,¹ Alexandra Funke,² Peter Hubwieser³

Abstract: Informatik soll in den kommenden Jahren immer stärker in die frühe Bildung integriert werden und somit den Weg in die Grundschulen sowie die Kindergärten finden. Dafür ist es erforderlich, Methoden und Inhalte zu erforschen, die für Schülerinnen und Schüler der Primarstufe geeignet sind. Auf dieser Basis haben wir einen dreitägigen Kurs, den Programmierzirkus, für Dritt- und Viertklässler entwickelt, der den Kindern einen Einblick in die Programmierung geben soll. Die Kinder sind zwischen acht und zehn Jahren alt und befinden sich kurz vor dem Übergang an die weiterführende Schule. Im Jahr 2016 nahmen 58 Schülerinnen und Schüler erfolgreich an dem Kurs teil. Weitere Durchführungen sind im Juni und Juli 2017 mit 63 Dritt- und Viertklässlern geplant. Dieser Beitrag beschreibt den Aufbau der einzelnen Kurstage und die verwendeten Lehr-Lern-Methoden. Weiterhin werden die Aufzeichnungen und Auswertungen des Programmierzirkus beschrieben, bei dem eine Vielzahl an Methoden zum Einsatz kam.

Keywords: Programmieren; informatische Bildung; Grundschule; CS unplugged; Scratch

1 Einleitung

Das Fach Informatik wird an Schulen und Universitäten mit verschiedenen Herausforderungen konfrontiert. Beispielsweise haben Schülerinnen und Schüler sowie Studentinnen und Studenten unterschiedliche Fehlvorstellungen und Vorurteile bezüglich des Faches [FBH16], die sich zum Großteil bereits in der Kindheit entwickeln und manifestieren [Pr14]. Um zu verhindern, dass die Lernenden solche meist negativen Einstellungen entwickeln, ist ein aktueller Forschungsansatz, informatische Konzepte wie Programmierung bereits in die Grundschule zu integrieren. Dies ermöglicht den Kindern, eigene Erfahrungen mit Informatik und neuen digitalen Medien zu sammeln. Sie lernen, den Computer nicht nur als Benutzer einzusetzen, sondern mit dessen Hilfe kreativ zu werden [AGE14]. Dieser Rollenwechsel in Kombination mit interessanten Erfahrungen könnte die Selbstwirksamkeit der Schülerinnen und Schüler im Allgemeinen und speziell bezüglich der Informatik und Programmierung erhöhen [To15]. Zeitgleich dazu steigt die Diskussion über die Notwendigkeit der Informatik in der frühen Bildung stetig. Während mehrere Länder bereits das Fach

¹ Technische Universität München, School of Education, Professur für Didaktik der Informatik, Arcisstr. 21, 80333 München, katharina.geldreich@tum.de

² Technische Universität München, School of Education, Professur für Didaktik der Informatik, Arcisstr. 21, 80333 München, alexandra.funke@tum.de

³ Technische Universität München, School of Education, Professur für Didaktik der Informatik, Arcisstr. 21, 80333 München, peter.hubwieser@tum.de

Informatik in die Grundschulcurricula integriert haben (z. B. Großbritannien [Br14] und Australien [FVF14]), hat Deutschland noch keine verbindlichen Richtlinien für den Umgang mit diesen Themen entwickelt. Um herauszufinden, welche Lehrmethoden und Inhalte für Kinder im Grundschulalter geeignet wären, haben wir einen einführenden Programmierkurs für Schülerinnen und Schüler der Grundschule entworfen.

2 Kurskonzept

Der Kurs wurde zunächst für die vierte Klasse konzipiert und hat das Ziel, den Schülerinnen und Schülern innerhalb von drei Tagen einen Einblick in die Programmierung zu geben. Dabei sollen sie ganz grundsätzlich verstehen, wie ein Computerprogramm funktioniert, und darüber hinaus befähigt werden, sich in der Programmierumgebung Scratch zurechtzufinden und dort eigene multimediale Projekte zu programmieren.

Das Thema „Zirkus“ zieht sich aus mehreren Gründen als durchgängiges Motiv durch den Kursablauf, die Aufgaben und die Materialien. Ein wichtiger Aspekt bei der Wahl dieses Themas war die Motivation der Schülerinnen und Schüler. Da die meisten Kinder persönliche Erfahrungen mit einem Zirkusbesuch verbinden können, ist die Themenwahl ein erster Schritt, um die Kursinhalte lebendig und interessant zu vermitteln und dabei an die Lebenswirklichkeit der Schülerinnen und Schüler anzuknüpfen. Des Weiteren hofften wir, Mädchen und Jungen mit diesem Thema gleichermaßen anzusprechen, da weder Mädchen noch Jungen vorrangig damit assoziiert werden. Zusätzlich bietet ein Zirkus vielfältige Aufgabenanlässe, in denen die Handlungen von Tieren und Personen simuliert werden können.

An jedem der drei Kurstage haben wir vier Stunden mit der Klasse verbracht, die nach und nach ein detaillierteres Bild vom Programmieren erhalten sollte. Am Ende des Kurses sollten alle Schülerinnen und Schüler dazu fähig sein, mit der Programmierumgebung Scratch zu arbeiten und algorithmische Kontrollstrukturen anzuwenden und zu kombinieren. Da man in der Grundschule prinzipiell mit einer hohen Variabilität von Schülerleistungen, Lernvoraussetzungen und Lerntypen konfrontiert ist, haben wir uns für eine Kombination verschiedener Lehr- und Arbeitsmethoden entschieden. Einen schematischen Überblick über den Ablauf, die Inhalte und die Ziele des Kurses zeigt Abb. 1.

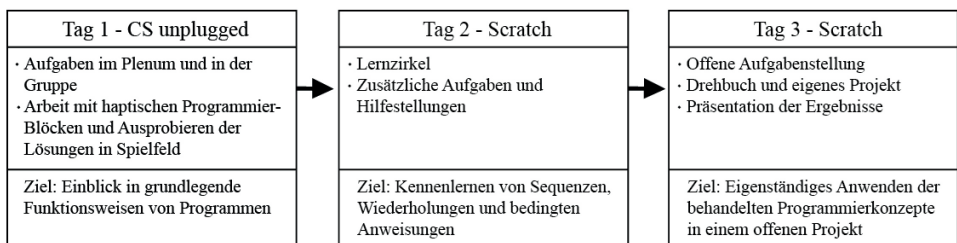


Abb. 1: Ablauf, Inhalte und Ziele des Kurses

Stets im Vordergrund stand durchgängig das Prinzip des aktiven Lernens [PI69] [Vo97]. Die Schülerinnen und Schüler sollten die Möglichkeit haben, sich eigentätig und aktiv mit den Lerngegenständen auseinanderzusetzen. Des Weiteren sollte die Möglichkeit gegeben sein, den informatischen Konzepten auf unterschiedliche Art und Weise zu begegnen [Br66].

2.1 Aufbau des Kurses

Tag 1. Da die meisten Schülerinnen und Schüler keinerlei Vorwissen im Bereich Programmierung oder allgemein in Informatik mitbrachten, war das Ziel des ersten Tages, ihnen eine Grundidee über die Funktionsweise eines Computerprogramms zu geben. Sie sollten begreifen, dass Programme eine bestimmte Aufgabe bewältigen, indem sie präzisen und klaren Anweisungen folgen. Um die Kinder nicht zu überfordern, wurden diese grundlegenden algorithmischen Konzepte „unplugged“ [CS17], das heißt ohne die Verwendung von Computern, eingeführt. Um zu lernen, komplexe Aufgaben in kleinere Teile zu zerlegen, bearbeiteten die Kinder eine Vielzahl von Übungen, in denen Vorgänge in eindeutige Anweisungen umgewandelt werden mussten. Verschiedene einfache Aufgaben wurden zunächst gemeinsam im Plenum gelöst, im Anschluss daran wurden verhältnismäßig komplexere Aufgaben in Kleingruppen bearbeitet. Dabei „programmierten“ die Schülerinnen und Schüler zuerst die Lehrkraft und im Anschluss daran sich gegenseitig. Um das Zirkusthema aufzugreifen, mussten in den Aufgaben beispielsweise vermisste Gegenstände gefunden oder entlaufene Tiere eingefangen werden. Sobald eine Aufgabe gelöst wurde, konnte man die Lösung in einem im Zimmer aufgebauten Spielfeld überprüfen (Abb. 2, rechts). Für die Bearbeitung der Aufgaben fertigten wir haptische Programmierblöcke an, die optisch den Programmierbefehlen in Scratch entsprechen (Abb. 2, links).



Abb. 2: Ein Schüler arbeitet mit den haptischen Scratchblöcken (links); die fertigen Programme werden im Parcours getestet (rechts)

Da die ausgedruckten und laminierten Blöcke mit Magneten und Klettverschlüssen versehen sind, kann mit ihnen sowohl an der Tafel als auch auf großen Filzbahnen gearbeitet werden. Zum einen konnten die Kinder auf diesem Wege ihre Programme ohne Probleme verändern

und im Raum transportieren, zum anderen erhofften wir uns, den Übergang zur Weiterarbeit in Scratch zu erleichtern.

Tag 2. Am zweiten Tag sollten die Schülerinnen und Schüler lernen, einfache Programme am Computer zu erstellen. Da wir eine kinderfreundliche Programmierumgebung verwenden wollten und um mögliche Frustrationen durch syntaktische Fehler zu vermeiden, entschieden wir uns für die visuelle Programmiersprache Scratch [Ma10]. Wir entwickelten einen Lernzirkel mit zunehmend schwierigen Stationen, in dem die Grundfunktionen von Scratch nacheinander behandelt wurden und die jedes Kind in seinem eigenen Tempo bearbeiten konnte. Ausgehend von Fragen der Bedienung führten die erstellten Zirkelkarten (Abb. 3) über einfachen Sequenzen hin zu Kontrollstrukturen wie Wiederholungen und bedingte Anweisungen. An jeder Station wurde die entsprechende Funktion zunächst schrittweise erklärt und die Schülerinnen und Schüler wiederholten jeden Schritt an ihrem eigenen Rechner. Im Anschluss daran bearbeiteten sie eine dazu passende Aufgabe, bei der die eingeführte Funktion in einem anderen Kontext oder leicht abgewandelt verwendet werden musste. Um die heterogene Gruppe gleichermaßen zu unterstützen, standen zusätzliche Aufgaben sowie hilfreiche Tipps für die komplexeren Aufgabenstellungen bereit.

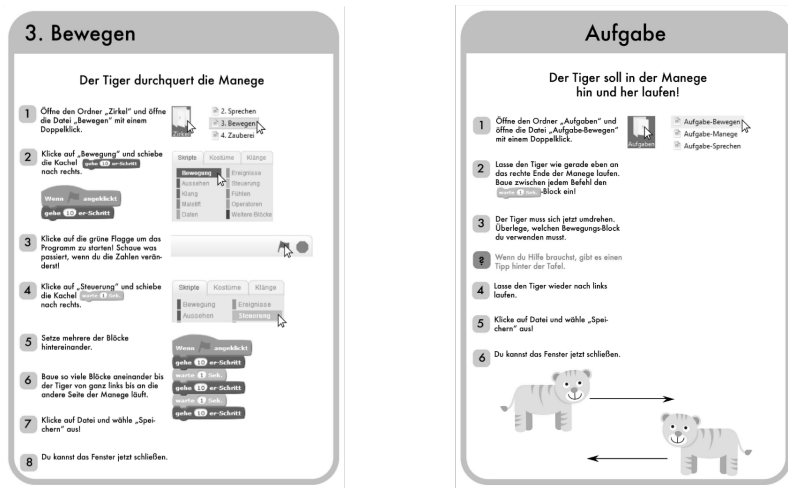


Abb. 3: Vorderseite (links) und Rückseite (rechts) einer Karte des Scratchzirkels

Tag 3. Ziel des dritten Tag war es, herauszufinden, was die Schülerinnen und Schüler gelernt hatten und ob sie das Gelernte auch in einer offeneren Aufgabe anwenden können. Darüber hinaus wollten wir den Kindern eine Möglichkeit bieten, kreativ und selbstbestimmt zu arbeiten. Jedes Kind sollte seine eigene Zirkusgeschichte in einem kurzen Drehbuch (Abb. 4, links) beschreiben und daraufhin in Scratch umsetzen. Da wir trotz der offenen Aufgabenstellung eine Vergleichbarkeit der Schülerprojekte gewährleisten wollten, legten wir einige Anforderungen fest, die erfüllt werden mussten. Die Programme sollten a) mehr als eine Figur beinhalten, b) die Figuren bei der Ausführung bewegen, c) mindestens eine

Wiederholung ausführen und d) mindestens eine bedingte Anweisung enthalten. Nach dem Erfüllen dieser Anforderungen konnten die Schülerinnen und Schüler ihre Arbeit ohne weitere Richtlinien fortsetzen. Ihnen stand es frei, mit Scratch zu experimentieren und ihre eigenen Zirkusgeschichten zu erfinden sowie diese im Anschluss umzusetzen (Abb. 4, rechts). Am Ende des Tages präsentierte jedes Kind seine Zirkusgeschichte der Klasse und hatte Gelegenheit, das Vorgehen und mögliche Schwierigkeiten zu kommentieren.

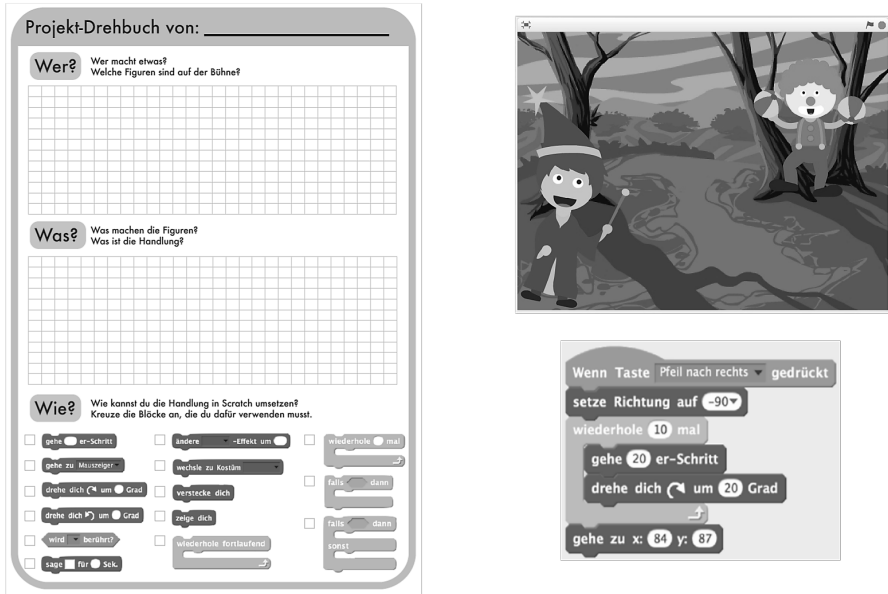


Abb. 4: Drehbuch für die Projekte (links) und das Projekt eines Kindes (rechts)

2.2 Aufzeichnung des Kurses

Um das Kurskonzept und die Effektivität des Kurses zu testen, wurden alle Kursdurchläufe mit einem Mixed-Methods Ansatz aufgezeichnet. Dafür kamen folgende Methoden kombiniert zum Einsatz:

Videografie. Um die Interaktionen der Kinder untereinander und mit der Lehrkraft sowie das Verhalten der Kinder allgemein zu untersuchen, wurden die kompletten Kurse mit vier Kameras aufgezeichnet. Für die entsprechende Tonaufzeichnung kamen ein Ansteckmikrofon für jede Lehrkraft und Richtmikrophone zum Einsatz.

Gruppeninterviews und Fragebögen. Wir nutzten eine Vielzahl an Interview- und Reflektionsmethoden, um einen Einblick in das Vorwissen der Schüler zu bekommen, darüber, welche Vorstellungen sie vom Programmieren haben und wie sie sich vor bzw. nach den Kurstagen fühlten.

Bildschirm- und Audioaufzeichnung. An den Kurstagen zwei und drei wurden die Bildschirme der Kinder mitgeschnitten. Die Sprache wurde über die internen Mikrophone der Laptops aufgenommen. Mit dieser Methode kann ein Einblick in die Arbeitsweise der Kinder erhalten werden.

Scratch-Ergebnisse. Alle Programmiererergebnisse der Kinder vom zweiten und dritten Tag wurden gespeichert, um sie im Nachhinein genauer analysieren zu können.

3 Beobachtungen und Auswertungen

Basierend auf der Vielzahl an Daten, die bei der Kursdurchführung erhoben wurden, führten wir verschiedene Auswertungen durch, um einen tieferen Einblick in die Arbeit der Kinder zu erlangen.

Mithilfe eines weiterentwickelten Code-Systems wurden alle 127 Scratch-Ergebnisse der Kinder des dritten Kurstages analysiert [FGH17]. Dabei fanden wir heraus, dass die Projekte vorrangig in drei Projekttypen unterteilt werden können: Spiel, Geschichte und Animation. Diese unterscheiden sich nicht nur in Art und Anzahl der verwendeten Blöcke, sondern auch in der Verwendung der Figuren im Programm. Annähernd alle Kinder erfüllten in ihren Projekten die von uns vorgegebenen Pflichtelemente (siehe 2.1 Tag 3).

Mit Hilfe von Reflektionsbögen, die am Ende jedes Kurstages von den Kindern ausgefüllt wurden, erhielten wir Einblicke, wie sie den Kurs empfanden. Die Bögen dienten zudem dem Erfassen des Vorwissens sowie den aufgetretenen Schwierigkeiten. Alles in allem hatte der Großteil der Kinder an allen Tagen viel Freude – vor allem an den Tagen, an denen mit den Computern gearbeitet wurde. Bei den CS unplugged Aufgaben empfanden die Kinder es als schwierig, den Kursleiter zu "programmieren". Doch nachdem diese erste Hürde gemeistert wurde, fiel den Kindern das Programmieren immer leichter. Dies lässt sich auch gut anhand der Videoaufzeichnungen beobachten. Am meisten Spaß hatten die Kinder am ersten Tag beim Ausprobieren der Lösungen im Spielfeld. Die Antworten des zweiten Tages waren deswegen für uns interessant, weil diese aufzeigen, welche Zirkelaufgaben besonders herausfordernd für die Kinder waren. Anhand dessen kann der Kurs entsprechend verändert werden. Alle Kinder würden gerne weiterhin in der Schule mit Scratch programmieren und etwa 90% wollen dies auch zuhause tun.

Die Bildschirm- und Tonaufzeichnungen der Laptops ermöglichen die Untersuchung des Vorgehens beim Programmieren. In Kombination mit den Videoaufzeichnungen lassen sich Stellen identifizieren, an denen die Kinder auf Probleme gestoßen sind. Diese sind bei den Kindern sehr individuell. Im nächsten Schritt folgt die Untersuchung, ob man verschiedene Programmierotypen unterscheiden kann.

4 Ausblick und zukünftige Arbeiten

Anhand der Beobachtungen und Auswertungen wurden die Materialien, das Kurskonzept und die Aufzeichnungsmethodik evaluiert und überarbeitet. Am Ende des Schuljahres 2016/17 werden mit dem überarbeiteten Kurskonzept vier weitere Durchläufe mit dritten und vierten Klassen durchgeführt. Anschließend sollen die erhaltenen Daten, wie am Beispiel der Daten aus 2016, ausgewertet und mit denen des Vorjahres verglichen werden.

Parallel arbeiten wir an einem Fortbildungskonzept, das Grundschullehrerinnen und -lehrern ohne einschlägiges Vorwissen ermöglichen soll, den Kurs an ihrer Schule zu halten. Fernziel ist hierbei, allen Viertklässlern des Landkreises Berchtesgadener Land die Teilnahme an einem Programmierkurs und somit das Erlernen grundlegender Programmierprinzipien zu ermöglichen.

5 Inhalte des Workshops

Nach einer kurzen thematischen Einführung in das Themengebiet „Programmieren in der Grundschule“ können sämtliche Kursmaterialien selbstständig ausprobiert werden. Darüber hinaus werden verschiedene Schülerergebnisse ausgestellt, anhand derer die Auswertung des Kurses aufgezeigt wird.

Literaturverzeichnis

- [AGE14] Armoni, Michal; Gal-Ezer, Judith: Early computing education. *ACM Inroads*, 5(4):54–59, 2014.
- [Br66] Bruner, Jerome: *Toward a theory of instruction*. Harvard University Press, Cambridge, 1966.
- [Br14] Brown, Neil C. C.; Sentance, Sue; Crick, Tom; Humphreys, Simon: Restart: the resurgence of computer science in UK schools. *ACM Transactions on Computing Education*, 14(2):1–22, 2014.
- [CS17] CSUnplugged.org: , Computer Science Unplugged. csunplugged.org, March 2017.
- [FBH16] Funke, Alexandra; Berges, Marc; Hubwieser, Peter: Different Perceptions of Computer Science. In: 2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE). IEEE, S. 14–18, 2016.
- [FGH17] Funke, Alexandra; Geldreich, Katharina; Hubwieser, Peter: Analysis of Scratch Projects of an Introductory Programming Course for Primary School Students. In: Proceedings of the 2017 IEEE Global Engineering Education Conference (EDUCON). IEEE, S. 1233–1240, 2017.
- [FVF14] Falkner, Katrina; Vivian, Rebecca; Falkner, Nickolas: The Australian Digital Technologies Curriculum: Challenge and Opportunity. In: Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148. ACE '14, Australian Computer Society, Inc, Darlinghurst, Australia, Australia, S. 3–12, 2014.

- [Ma10] Maloney, John; Resnick, Mitchel; Rusk, Natalie; Silverman, Brian; Eastmond, Evelyn: The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4):16:1–16:15, 2010.
- [PI69] Piaget, Jean; Inhelder, Bärbel: *The psychology of the child*. Basic Books, 1969.
- [Pr14] Prottzman, Kiki: Computer science for the elementary classroom. *ACM Inroads*, 5(4):60–63, 2014.
- [To15] Topi, Heikki: Gender imbalance in computing. *ACM Inroads*, 6(4):22–23, 2015.
- [Vo97] Vollmers, Burkhard: Learning by doing - Piagets konstruktivistische Lerntheorie und ihre Konsequenzen für die pädagogische Praxis. *International Review of Education*, 43:73–85, 1997.

Der Hamster hat das Programm erfolgreich beendet – Grundschul Kinder lernen Programmieren

Otto Thiele¹, Petra Jückstock²

Abstract: Seit 2014 führten die Autoren des Praxisbeitrags jedes Schuljahr eine Arbeitsgemeinschaft Informatik für Kinder der Klassenstufe vier durch. Die Kinder wurden mittels des Hamster-Struktogrammeditors „HaSE“ an das Programmieren herangeführt. Dies erfolgte, indem die Kinder für die Probleme des Hamsters Lösungsalgorithmen entwickelten und als Hamster-Struktogramme implementierten. Da die Struktogramme als Programme interaktiv visuell ablaufen, konnten die Kinder testen, ob der Hamster ausführte, was sie sich vorgestellt und gedacht hatten. Begleitet wurde das Lernen der Kinder unter Berücksichtigung des lerntheoretischen Ansatzes „Verständnisintensives Lernen“ und des professionstheoretischen Ansatzes „Verstehen zweiter Ordnung“.

Keywords: Arbeitsgemeinschaft Informatik für Viertklässler, Algorithmen entwickeln, Hamster-Struktogramme, Hamster-Struktogrammeditor „HaSE“, Programmieren, „Verständnisintensives Lernen“, „Verstehen zweiter Ordnung“

1 Die Arbeitsgemeinschaft

Anfang des Jahres 2014 wurde das Konzept zur Durchführung einer schulübergreifenden Arbeitsgemeinschaft Informatik für Jenaer Kinder der vierten Klassen erstellt. Danach erfolgte die Ausschreibung.

„Mein Name ist Bommel³. Ich bin ein kleiner Hamster. Kein richtiger Hamster – denn ich lebe nicht in der Natur und auch nicht in einem Käfig mit Laufrad. Am liebsten hamstere ich nach Herzenslust in Fantasiewelten des Computers. Diese Welten kannst Du Dir für mich ausdenken und am Computer gestalten. Mir wird es dort bestimmt gefallen. Wenn Du auch noch meine Sprache lernst, dann werde ich auf Dich hören und mit Freude und Spaß ausführen, was Du mir sagst.“ [wi16]

Für die Eltern wurde in der Ausschreibung ergänzt, dass das Konzept der Arbeitsgemeinschaft darauf ausgerichtet ist, das analytische, logische und algorithmische Denken der Kinder altersgerecht und spielerisch anzuregen und zu fördern. Das heißt, die Kinder sollen befähigt werden, algorithmische Probleme am Computer mit dem an der Universität Oldenburg entwickelten Hamster-Struktogrammeditor⁴ „HaSE“ zu lösen [wi16].

¹ Verständnisintensives Lernen e. V., Löbstedter Straße 67, 07749 Jena, otto.thiele@verstehenlernen.de

² witelo e. V., Löbstedter Straße 67, 07749 Jena, p.jueckstock@witelo.de

³ Für die Kinder war es wichtig, dass der Hamster einen Namen hatte.

⁴ Der Hamster-Struktogrammeditor „HaSE“ verfügt über eine für die Kinder leicht bedienbare grafische Benutzeroberfläche, die aus der gestaltbaren virtuellen Welt des Hamsters und dem Editor für Hamster-Struktogramme besteht [Bo16].

Die Resonanz auf die Ausschreibung war groß, so dass die Arbeitsgemeinschaft im Schuljahr 2014/15 erstmals durchgeführt werden konnte. Wie die Kinder seitdem lernten und beim Lernen begleitet wurden, wird nachfolgend beschrieben.

2 Mit dem Hamster lernen

In der Arbeitsgemeinschaft waren die Aufgaben in Geschichten verpackt, die die Fantasie der Kinder anregten. Eine solche Aufgabe war beispielsweise die vom „Hamster Bommel als Nachtwächter“.

Jena war im Mittelalter eine kleine Stadt (Abb. 1) mit einer Stadtmauer und vier Stadt-toren. Die Stadtmauer war der Schutz vor räuberischen Angriffen. Das Johannistor, der Pulverturm und Teile des Anatomieturms sowie der Stadtmauer sind bis heute erhalten geblieben.

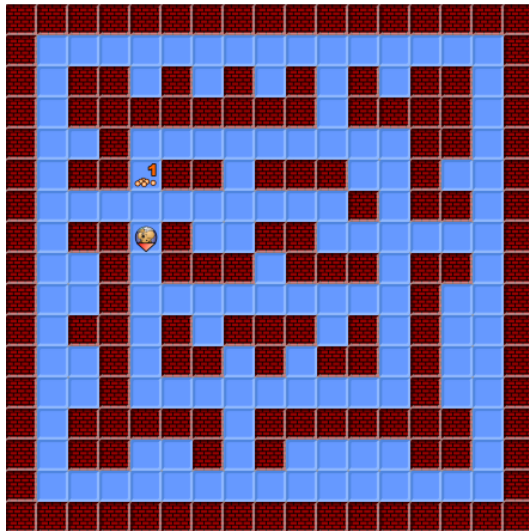


Abb. 1: Das mittelalterliche Jena mit Hamster Bommel

Jena hatte damals immer einen Nachtwächter. Der Wächter lief nachts innerhalb von Jena die Stadtmauer ab. Ein Zauberer hat Bommel in das mittelalterliche Jena gebeamt. Dort muss Bommel die Aufgabe des Nachtwächters übernehmen und die Stadtmauer ablaufen. Als Belohnung darf er nach dem Rundgang das leckere Korn verspeisen.

Um diese und andere Aufgaben lösen zu können, lernten die Kinder Algorithmen zu entwickeln. Dazu eigneten sie sich die Hamster-Befehle, Testbefehle, zusammengesetzten Bedingungen, Grundstrukturen (Sequenz, Selektion, Zyklus) und parameterlose Prozeduren an [Bo16].

2.1 Lösen der Einstiegsaufgabe „Hamster Bommel und das leckere Korn“

Bommel schaut aus seinem Bau aufs Feld (Abb. 2). Plötzlich erblickt er am Feldrand ein leckeres Korn. Bommel wäre kein Hamster, würde er das Korn nicht hamstern wollen.

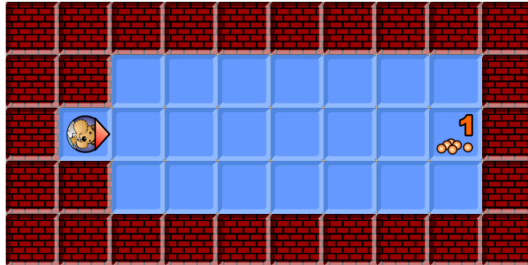


Abb. 2: Die Welt des Hamsters Bommel

- Baue Bommels Welt am Computer.
- Hilf Bommel, das Korn zu hamstern.
- Prüfe, ob Bommel die Aufgabe löst.

Als hilfreich erwies sich, dass die Kinder die Aufgabe zunächst in einem Rollenspiel auf den Fliesen zwischen den Wänden des Schulfloors lösten (Abb.3).



Abb. 3: Erfahrungen gehören zum Lernen (Foto: Thomas Corbus)

Im Spiel übernahm jedes Kind die Rolle des Hamsters. Dabei galt es zu beachten, dass der Hamster beim Laufen jeden Schritt nur von einer zur nächsten Fliese ausführen kann. Obwohl alle Kinder den kürzesten Weg zur gegenüberliegenden Wand beschrritten, benötigten sie unterschiedlich viele Schritte. Über die Ursachen nachdenkend fanden sie heraus, dass manche von ihnen die Schritte nicht immer „hamsterkonform“ ausführten. Das lag zum einen daran, dass es schwierig war, Fliese für Fliese in kleinen Schritten zu durchlaufen und zum anderen an mangelnder Konzentration. Beim Wiederholen achteten die Kinder untereinander darauf, dass alle mit gleich vielen Schritten zur Wand gelangten. An der Wand angekommen, nahm jedes Kind das Korn in Form eines Papierschnipsels auf. Mit dem korrekten Ausführen der Handlungsfolge, Schritt für Schritt bis zur Wand zu laufen

und das Korn aufzunehmen, hatten die Kinder die Aufgabe im Rollenspiel gelöst. Zum Rollenspiel sei bemerkt, dass die Kinder insbesondere durch das eigene Handeln und Erleben lernten. Das Rollenspiel hat sich zudem als geeignete Methode erwiesen, die Kinder an die Hamster-Befehle und Grundstrukturen der Algorithmen heranzuführen.

Nach einer kurzen Einführung war es den Kindern problemlos möglich, die virtuelle Welt des Hamsters am Computer zu bauen. Danach ging es damit weiter, einen Algorithmus zu entwickeln. Dazu überlegten sich die Kinder, dass der Hamster in der virtuellen Welt, analog zum Rollenspiel, eine Folge von Schritten ausführen muss, um von seinem Bau zur gegenüberliegenden Wand zu gelangen. Es ergab sich die Frage, wie der Hamster am Computer zum Laufen gebracht werden kann. Als Antwort auf die Frage lernten die Kinder die Notation `vor()` des Hamster-Befehls, sich einen Schritt vor zu bewegen, und die Darstellung im Hamster-Struktogramm kennen. Im Rollenspiel entsprach `vor()`, genau einen Schritt vor zur nächsten Fliese zu laufen. Letztlich mussten die Kinder noch wissen, wie der Hamster am Computer ein Korn „hamstern“ kann. Dazu lernten sie die Notation `nimm()` des Hamster-Befehls, ein Korn aufzunehmen, und die Darstellung im Hamster-Struktogramm kennen. Mit den Erfahrungen aus dem Rollenspiel, der Kenntnis der Hamster-Befehle und deren Notationen entwickelten die Kinder Handlungsfolgen zum Lösen der Aufgabe am Computer (Abb. 4).



Abb. 4: Ein Hamster-Struktogramm erstellen (Foto: Thomas Corbus)

Die Handlungsfolgen stellten sie als Struktogramme (Abb. 5) mithilfe des Hamster-Struktogrammeditors dar. Da Hamster-Struktogramme auf grafische Weise Hamster-Programme repräsentieren, die interaktiv ausführbar sind, konnten die Kinder überprüfen, ob der Hamster tatsächlich von seinem Bau zur gegenüberliegenden Wand läuft und das Korn „hamstert“.

Es kam vor, dass Hamster-Struktogramme mancher Kinder nicht zu den von ihnen erwarteten Programmabläufen führten. Beispielsweise, wenn der Hamster die Wand nicht erreichte, weil er zu wenige Schritte ausführte und die Fehlermeldung „Hamster versucht ein Korn aufzunehmen, das nicht da ist. :-“ ausgegeben wurde oder er zu viele Schritte ausführen sollte und die Fehlermeldung „Hamster läuft gegen eine Mauer! :-“ ausgegeben wurde. Solche Erfahrungen stimmten die Kinder nachdenklich. Sie suchten die Ursachen, weshalb der Hamster nicht tat, was sie sich vorgestellt hatten.

```

void main()
  vor();
  vor();
  vor();
  vor();
  vor();
  vor();
  vor();
  nimm();
    
```

Abb. 5: Algorithmus aus einer Sequenz von Hamster-Befehlen

Beginnend mit dem Rollenspiel, dem Bauen der Hamster-Welt, dem Erstellen der Hamster-Struktogramme am Computer sowie dem Testen des Programmablaufs und den sich eventuell daraus ergebenden Nachbesserungen gelang es allen Kindern, die Aufgabe zu lösen. Das Lösen belohnte das Programm mit der Ausgabe „Hamsterprogramm erfolgreich beendet! Herzlichen Glückwunsch, der Hamster hat das Hamsterprogramm erfolgreich beendet. :-)“.

Das Hamster-Struktogramm ließ sich durch das Einführen der while-Anweisung und des Hamster-Testbefehls `vornFrei()` modifizieren (Abb. 6).

```

void main()
  vornFrei() ?
  vor();
  nimm();
    
```

Abb. 6: Modifiziertes Hamster-Struktogramm

Bevor die Kinder allerdings das Struktogramm mit dem Hamster-Struktogrammeditor modifizierten, führten sie auf den Fliesen des Schulflurs im Rollenspiel die Handlungsfolge „solange vorn frei, laufe Schritt für Schritt vor und hamstere danach das Korn“ aus.

Die Aufgabenstellung wurde darüber hinaus noch dadurch erweitert, dass der Hamster mit dem Korn in seinen Bau zurückkehren, das Korn dort ablegen und wieder aus dem Bau schauen sollte. Das Lösen dieser und weiterer Aufgaben sowie das damit verbundene Lernen führte die Kinder zu einem „intuitiven Algorithmusbegriff“, der sich auf Handlungsabläufe bezieht, die als Programme auf dem Computer ausführbar sind.

Nach und nach zeigte sich, dass die Kinder beim Entwickeln von Hamster-Struktogrammen nicht mehr auf Rollenspiele angewiesen waren, weil sie sich die vom Hamster auszuführenden Handlungen immer besser vorstellen konnten.

2.2 Lösen fortgeschrittener Aufgaben

Die Aufgabe „Hamster Bommel als Nachtwächter“ (Abb. 1 in Abschn. 2) lösten die Kinder aufgrund der Erfahrungen, die sie bereits in den Rollenspielen und beim Entwickeln von Algorithmen gemacht hatten. Sie überlegten sich, dass der Hamster entlang der Stadtmauer zum Korn gelangt, indem im Hamster-Struktogramm viermal nacheinander die while-Anweisung und der Befehl `linksUm()` (Abb. 7) angeordnet und abschließend noch die Befehle `vor()` und `nimm()` anfügt werden. Am Computer verfolgten die Kinder mit, wie das Programm ablief und der Hamster entlang der Stadtmauer zum Korn gelangte und es aufnahm. Damit war die Aufgabe gelöst.



Abb. 7: Schleife gefolgt von einem Hamster-Befehl

Leistungsstarke Kinder wurden angeregt, sich weiter mit der Aufgabe zu beschäftigen. Mit Hilfestellungen fanden sie heraus, dass das Wiederholen der while-Anweisung und des Befehls `linksUm()` im Hamster-Struktogramm durch eine äußere Schleife realisiert werden kann. Allerdings ergab sich dabei beispielsweise das Problem, dass der Hamster Runde für Runde lief, nicht zum Stehen kam und dadurch auch nicht seine Belohnung bekam. Jetzt wurde es schwierig, denn die Frage ergab sich, wie der Hamster zum Stehenbleiben gebracht werden kann. In diesem Zusammenhang lernten die Kinder die zusammengesetzte Bedingung `!(kornDa())` kennen (Abb. 8).

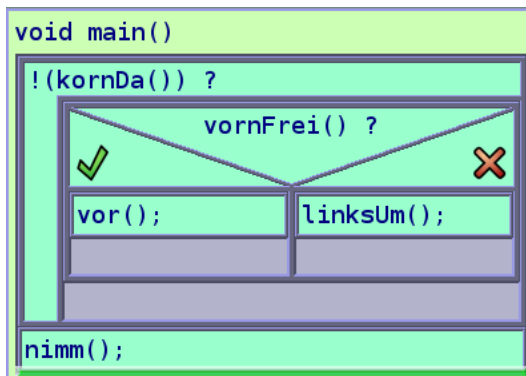


Abb. 8: Lösungsalgorithmus mit hohem Anspruch

Im Laufe der Zeit lernten die Kinder immer besser, analytisch und algorithmisch zu denken. Sie waren in der Lage, in den umrahmenden Geschichten der Aufgaben die Probleme zu erfassen und Lösungsalgorithmen zu entwickeln. Ebenso waren einige Kinder in der Lage, gegebene Aufgaben zu erweitern oder zu modifizieren.

Spaß und Freude bereitete den Kindern, sich zu gegebenen Hamster-Welten (Abb. 9) Geschichten und Aufgabenstellungen auszudenken, wie z. B. die folgende Geschichte.

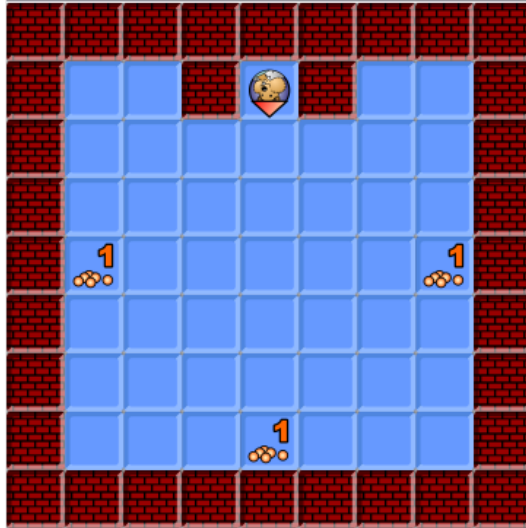


Abb. 9: Vorgegebene Hamster-Welt

„Heute hat der kleine Hamster Bommel Lust auf einen Spaziergang. Er steckt seinen Kopf aus dem Bau. Da erblickt er die drei Körner, die verteilt auf dem Feld rumliegen. Hamster Bommel folgt den Körnern. Nehmen tut er sie aber nicht. Bommel lässt sie lieber für die Vögel übrig. Am Ende sitzt der kleine Hamster Bommel wieder in seinem Bau, als ob nichts gewesen wäre. :-)“

Die Kinder dachten sich auch eigene fantasievolle Hamster-Welten und dazugehörige Aufgaben aus, die sie in Geschichten einbanden.

Außerdem lernten die Kinder parameterlose Prozeduren kennen und wendeten sie an. Beispielsweise erwies es sich als zweckdienlich, mithilfe einer Prozedur aus drei Hamster-Befehlen `linksUm()` den Befehl `rechtsUm()` zu erstellen.

```
void rechtsUm()  
{  
    linksUm();  
    linksUm();  
    linksUm();  
}
```

Abb. 10: Eine parameterlose Prozedur

Nach einem halben Jahr konnten die Kinder bereits anspruchsvolle Aufgaben mit dem Hamster-Struktogrammeditor lösen. Beispielsweise entwickelten sie Algorithmen, mit denen der Hamster aus gegebenen Bildnegativen (Abb. 11) die Bildpositive machte. Natürlich bot es sich an, dass die Kinder ihre Vornamen in Bildnegativen versteckten. Dann sollte der Hamster die Vornamen schreiben.

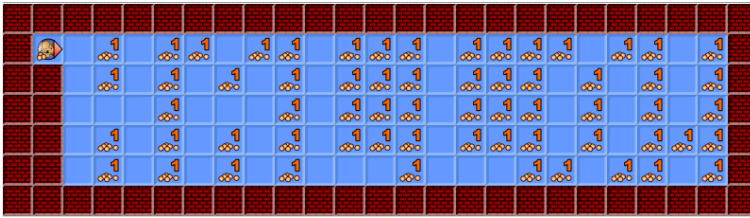


Abb. 11: Aus dem Bildnegativ das -positiv erstellen

3 Das Lernen begleiten

3.1 Verständnisintensives Lernen

Bei der pädagogischen Arbeit mit den Kindern in der Arbeitsgemeinschaft wurde der lerntheoretische Ansatz „Verständnisintensives Lernen“ [Fa03, Fa08] berücksichtigt.

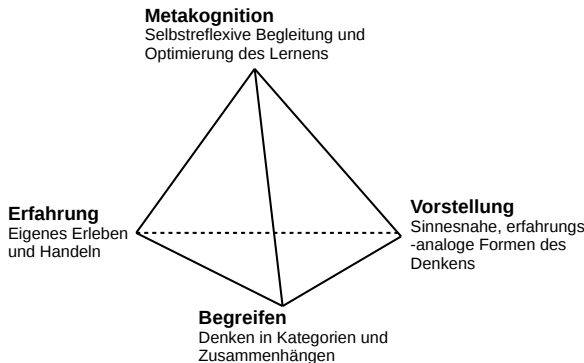


Abb. 12: Tetraedermodell des „Verständnisintensiven Lernens“ [Vi17]

Der Ansatz geht davon aus, dass die Qualität des Lernens zum einen durch das Verstehen bestimmt wird. Verstehen ist grundlegend für das Lernen und lässt sich als kognitiver Modellierungsprozess beschreiben, in dem Erfahrung, Vorstellung, Begreifen⁵ und Metakognition dynamisch zusammenwirken. In der Abb. 12 sind die vier Dimensionen des Verstehens grafisch veranschaulicht⁶. Zum anderen bestimmen die Autonomie, soziale Eingebundenheit und Kompetenzen der Lernenden die Lernqualität [Fa08, DR93].

Der lerntheoretische Ansatz erwies sich als hilfreich, die Verstehensprozesse der Kinder mitverfolgen, die individuellen Unterschiede beim Lernen genauer erfassen und besser berücksichtigen sowie die Ursachen für Verstehensprobleme analysieren zu können. Dazu einige erläuternde Beispiele.

⁵ Begreifen wird im lerntheoretischen Ansatz als eine Dimension des Verstehens aufgefasst.

⁶ Erfahrung, Vorstellung und Begreifen sind dem Denken und Metakognition ist dem Nachdenken zuordbar.

Wie wichtig eigenes Erleben und Handeln beim Lernen waren, zeigte sich u. a. in den Rollenspielen, die mit den Kindern im Rahmen des Lösen von Aufgaben durchgeführt wurden (Abschn. 2.1).

Interessant war zu beobachten, wie die Kinder erlebten, was der Hamster bei den Programmabläufen ausführte oder auch nicht (Abschn. 2.1). Führte der Hamster nicht das Erwartete aus, wurden die Kinder angeregt, über ihre Algorithmen nachzudenken, nach den Ursachen, weshalb die Algorithmen nicht das Erwartete leisteten, zu suchen und die Algorithmen nachzubessern.

Verstehensprobleme ergaben sich beispielsweise, wenn in Wortlauten von Aufgaben Begriffe verwendet wurden (wie etwa Feldrain), deren Bedeutung die Kinder nicht kannten oder mit denen sie andere Vorstellungen verbanden. Deshalb mussten Formulierungen von Aufgaben geändert werden.

Das logische Denken der Kinder war u. a. bei den zusammengesetzten Bedingungen gefordert. Formale Notationen, wie z. B. `!vornFrei()` oder `vornFrei() && !kornDa()`, bereiteten Probleme und mussten insbesondere geübt werden.

Die Kinder lernten selbstbestimmt („aus sich selbst heraus“) aus Interesse an den Geschichten und den vom Hamster zu bewältigenden Problemen. Wenn Kinder einmal nicht lernten, am Computer in der Welt des Hamsters versanken und keine Probleme lösen wollten, lag es oft am anstrengenden Schultag, den sie bereits hinter sich hatten. Dann ließ man diese Kinder auch einmal in der Welt des Hamsters spielen.

Die Kinder fühlten sich stets in die Arbeitsgemeinschaft eingebunden. Zu keiner Zeit fehlte es einem Kind an Akzeptanz bzw. Anerkennung innerhalb der Lerngemeinschaft.

Das auf Verstehen ausgerichtete Lernen trug mit dazu bei, dass die Kinder folgende fachliche Kompetenzen erwarben:

- Analysieren von Aufgaben, um zu lösende Probleme zu erfassen
- Entwickeln von Lösungsalgorithmen unterschiedlicher Schwierigkeitsgrade
- Darstellen der Lösungsalgorithmen in Hamster-Struktogrammen
- Suchen von Fehlern und Korrigieren fehlerbehafteter Algorithmen
- Lesen und Verstehen gegebener Hamster-Struktogramme
- Erweitern und Modifizieren gegebener Aufgaben
- Erstellen und Verwenden von parameterlosen Prozeduren
- Konstruieren von Aufgaben, Schreiben umrahmender Geschichten für diese Aufgaben und Erstellen von Musterlösungen

3.2 Verstehen zweiter Ordnung

Neben dem lerntheoretischen Ansatz fand der professionstheoretischen Ansatz „Verstehen zweiter Ordnung“ [Fa08] Berücksichtigung. Das heißt, es wurde ein professionell kontrollierter Perspektivwechsel zum Lernen der Kinder vollzogen, der ein ko-konstruierendes

Begleiten des Lernens ermöglichte. Durch Ko-Konstruktion fand ein Lernen statt, das durch Interaktion und Zusammenarbeit mit den Kindern bestimmt war. Beispielsweise wurden beim Ko-Konstruieren immer wieder Differenzenerfahrungen bzgl. des eigenen Verstehens und dem der Kinder gemacht, wenn Kinder Aufgaben und Probleme auf unerwartete und teils bemerkenswerte Art und Weise lösten. Diese Erfahrungen waren ausschlaggebend dafür, den Kindern die Denkwege offen zu halten. Führten die Gedankengänge bei Kindern zu inkorrekten oder keinen Lösungen, erwies sich Ko-Konstruktion ebenso als helfend. Dabei wurde in der Interaktion mit diesen Kindern anerkannt, dass sie sich bei ihren Lösungsversuchen etwas dachten. Was sie sich gedacht hatten, ließ man sich von ihnen erklären. Keinesfalls wurde gegenüber den Kindern geäußert, dass das, was sie gemacht hatten, falsch sei. Stattdessen wurden die Kinder angeregt, über ihre Gedankengänge nachzudenken und Korrekturen vorzunehmen oder andere Wege hin zu den Lösungen einzuschlagen.

4 Fazit

Ein Journalist der Ostthüringer Zeitung, der über die Arbeitsgemeinschaft den Artikel „Informatik kommt bei Viertklässlern gut an“ verfasste, stellte die Frage, ob Grundschulkinder überhaupt die Voraussetzungen mitbringen, sich mit informatischen Problemen zu befassen [Zi15]. Die Frage kann aufgrund der guten Erfahrungen aus der Arbeitsgemeinschaft, der erzielten Lernergebnisse und -qualität sowie dem Spaß und der Freude der Kinder beim Lernen bejaht werden.

Literaturverzeichnis

- [Bo16] Boles, D.: HaSE-Hamster-Struktogrammeditor. <http://www.boles.de/hamster/hase.html>, 2016. Stand: 09.03.2017.
- [DR93] Deci, E.; Ryan, R.: Die Selbstbestimmungstheorie der Motivation und ihre Bedeutung für die Pädagogik. *Zeitschrift für Pädagogik* 39, S. 223–237, 1993.
- [Fa03] Fauser, P.: Lernen als innere Wirklichkeit. Über Imagination, Lernen und Verstehen. In (Rentsch, I.; Madelung, E.; Fauser, P., Hrsg.): *Bilder im Kopf. Texte zum Imaginativen Lernen*, S. 201–213. Kallmeyer, Seelze-Velber, 2003.
- [Fa08] Fauser, P.: Theorie des Verständnisintensiven Lernens. In (Fauser, P.; Prenzel, M.; Schratz, M., Hrsg.): *Was für Schulen! Profile, Konzepte und Dynamik guter Schulen in Deutschland*, S. 5–26. Kallmeyer, Seelze-Velber, 2008.
- [Vi17] ViL e.V.: Das Tetraeder im ViL. http://www.verstehenlernen.de/?page_id=2788, 2017. Stand: 09.03.2017.
- [wi16] witelo e.V.: Hamster Bommel und der Computer. <http://witelo.de/lernorte/projekt/hamster-bommel-und-der-computer>, 2016. Stand: 10.03.2017.
- [Zi15] Zippel, T.: Informatik kommt bei Viertklässlern gut an. *Ostthüringer Zeitung*, 14. Februar 2015.

Workshop: Ein Durchgang durch das Grundfach Informatik (nur) mit dem elektronischen Schulbuch inf-schule.de

Heiko Jochum¹, Klaus Becker², Martin Zimnol³, Daniel Jonietz⁴, Oliver Schneider⁵,
Manuel Froitzheim⁶

Abstract: Anhand eines möglichen Durchgangs durch das Grundfach Informatik wird in einem ersten Teil ein Überblick über die Inhalte des elektronischen Schulbuchs inf-schule.de vermittelt. Der Umfang des Schulbuchs ist dabei so groß, dass er alle Inhalte des Grundfachs (zumindest in einigen Bundesländern) abdeckt. Innerhalb des Workshops wird neben diesen Inhalten auch ein Ausblick auf alternative Herangehensweisen und optionale Inhalte gegeben. In einem zweiten Teil entwickeln die Teilnehmenden einen möglichen Durchgang durch das Grundfach Informatik für ihr Bundesland. Die Teilnehmenden benötigen nur ein Endgerät mit Internetzugang.

Keywords: Elektronisches Schulbuch, Digitales Schulbuch, Informatikunterricht, Lernen, Didaktik der Informatik, Web-Applikation, Grundfach

1 Das Schulbuchprojekt inf-schule

Das Schulbuchprojekt inf-schule.de (vgl. <https://www.inf-schule.de/>) wurde 2008 initiiert und inzwischen ist ein elektronisches Schulbuch für den Informatikunterricht entwickelt worden, das fast die gesamten Unterrichtsinhalte des Faches Informatik abdeckt. Bezüglich des Umfangs (mit mehr als 2000 Webseiten) und der Nutzerzahlen (mit mehr als 100.000 Zugriffen pro Schulwoche) hat sich inf-schule.de inzwischen zum akzeptierten Lehrwerk für den Informatikunterricht entwickelt. Genau wie gedruckte Schulbücher orientiert es sich an den Lehrplänen und den Bildungsstandards und bietet vielfältiges Material zum Lernen und Lehren. ([F+17], ebenfalls in diesem Band)

Das Schulbuch inf-schule.de wird als digitales Schulbuch im Internet angeboten. In der nach Sanguo, Xuehai und Chenglin [Sa12] vorgeschlagen Entwicklungshierarchie für digitale Schulbücher ist es der Stufe 2.0 (Schulbücher ohne analoge Vorlage) zuzurechnen. In weiten Teilen erreicht es inzwischen auch die Stufe 3.0, die eine Mitgestaltung durch die Nutzer vorsieht.

¹ Werner-Heisenberg-Gymnasium Bad Dürkheim, Kanalstr. 19, 67098 Bad Dürkheim, jom@whgonline.de

² BurgGymnasium Kaiserslautern, Burgstraße 18, 67659 Kaiserslautern, beckerkp@web.de

³ Pädagogisches Landesinstitut Rheinland-Pfalz, Butenschönstraße 2, 67346 Speyer, martin.zimnol@pl.rlp.de

⁴ BurgGymnasium Kaiserslautern, Burgstraße 18, 67659 Kaiserslautern, daniel@jonietz.de

⁵ Sickingen-Gymnasium Landstuhl, Philipp-Fauth-Straße 3, 66849 Landstuhl, o.schneider@mac.com

⁶ Universität Siegen, Zentrum für ökonomische Bildung in Siegen (ZöBiS), Kohlbettstraße 15, 57068 Siegen, froitzheim@zoebis.de

2 Workshopbeschreibung

Durch den recht großen Umfang des Schulbuchs inf-schule.de können nahezu alle Unterrichtsinhalte des Grund- und auch des Leistungsfachs Informatik abgedeckt werden. Darüber hinaus gibt es etliche optionale Themenbereiche (wie z.B. Messwert-Erfassung mit dem RaspberryPi), die zusätzlich in den Unterricht integriert werden können.



Abbildung 1: Startseite mit der Übersicht über die verschiedenen Themenbereiche auf inf-schule.de (vgl. <https://inf-schule.de/>)

Innerhalb des Workshops wird ein Überblick über den Aufbau und die verschiedenen Themenbereiche vermittelt. Dazu soll zunächst ein möglicher Durchgang durch das Grundfach Informatik skizziert werden, der vollständig durch das Schulbuch inf-schule.de abgedeckt wird und dem rheinland-pfälzischen Lehrplan entspricht. Des Weiteren werden die darüber hinausgehenden Themenbereiche skizziert.

Durch die Vielzahl der zusätzlichen Optionen innerhalb des Schulbuchs sind verschiedene Wege durch das Schulbuch denkbar. Deshalb sollen in einem zweiten Teil die Teilnehmenden einen eigenen Durchgang durch das Grundfach Informatik für ihr Bundesland erarbeiten und überprüfen, inwieweit dieser sinnvoll durch das Schulbuch inf-schule.de abgedeckt wird.

Da es sich bei inf-schule.de um ein kostenloses online-Schulbuch handelt, benötigen die Teilnehmenden nur ein Endgerät mit Internetzugang.

Literaturverzeichnis

- [Da16] Dagstuhl-Erklärung: Bildung in der digitalen vernetzten Welt, www.gi.de/fileadmin/redaktion/Themen/dagstuhl-erklaerung-bildung-in-der-digitalen-welt-2016.pdf, Stand: 23.01.2017.
- [F+17] Froitzheim, M.; Becker, K.; Zimmol, M.; Jonietz, D.; Schneider, O.; Jochum, Heiko: Förderung digitaler Bildung mit digitalen Schulbüchern. (ebenfalls in diesem Band)
- [Sa12] Sanguo, C.; Xuehai, M.; Schenglin, L. (2012). The Pace of Ebook Development in China. IN: LOGOS: The Journal of the World Book Community, 23(2), 14-20.
- [ScFr14] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [ScFr15] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch 2015. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [ScFr16] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch 2016. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [Wi03] Wiater, W. (2003): Schulbuchforschung in Europa. Bestandsaufnahme und Zukunftsperspektive. Bad Heilbrunn: Klinkhardt.

Analogien für Programmierkonzepte: Ein Weg zum Computational Thinking

Barbara Wieczorek¹, Liz Ribe², Christina B. Class³ und Michael Brinkmeier⁴

Abstract: Im Rahmen einer Informatikausbildung ist das Erlernen einer Programmiersprache für Schüler und Studenten ein wichtiger Zugang, um Computational Thinking zu erlernen. Jedoch stellen die Grundlagen der Programmierung häufig eine schwer überwindbare Hürde dar, die Lernende vom eigentlichen Ziel abhält, Programmierkonzepte als Problemlöseoperatoren einzusetzen. In diesem Beitrag wird untersucht, inwieweit Analogien als Brücke zwischen dem Vorwissen des Lernenden und Programmierkonzepten dienen können. Es wird ein anschauliches Modell zur Verdeutlichung von Beziehungen und Wechselwirkungen von Lerner, Analogien und Fachkonzepten vorgestellt. In einem Workshop werden in Arbeitsgruppen Praxisbeispiele von Analogien für konkrete Programmierkonzepte zusammengestellt und deren Vor- und Nachteile sowie Grenzen und resultierende Herausforderungen diskutiert.

Keywords: Computational Thinking, Programmierung, Informatikunterricht, Analogien, Problemlösen, duale Kodierung

1 Einleitung

Computational Thinking wurde bereits in den 1950er und 60er Jahren als Teil der Informatik, unter dem Begriff *algorithmic thinking* (Algorithmisches Denken) gesehen [De09]. 2006 forderte Jeannette Wing, dass Computational Thinking neben Lesen, Schreiben und Rechnen als grundlegende Fähigkeit erlernt werden soll [Wi06]. Hierunter versteht sie eine Reihe von mentalen Werkzeugen, um Probleme zu lösen, Systeme zu entwerfen und menschliches Verhalten zu verstehen [Wi06]. Grundlagen des Computational Thinking sollen daher in die Curricula der Schule integriert werden [BS11]. Hierzu wurde eine Matrix mit Konzepten und verschiedenen Schulfächern entwickelt, in denen diese Konzepte angewandt und trainiert werden. Auch wenn Aspekte des Computational Thinking auf verschiedenste Wege gelehrt werden können, ist es zutiefst mit Programmieraktivitäten verbunden.

Das Erlernen einer Programmiersprache kann die Ausbildung von Computational-

¹ Ernst-Abbe-Hochschule Jena, Grundlagenwissenschaften, Carl-Zeiss-Promenade 2, 07745, Jena, Barbara.Wieczorek@eah-jena.de

² Ernst-Abbe-Hochschule Jena, Grundlagenwissenschaften, Carl-Zeiss-Promenade 2, 07745, Jena, Elizabeth.Ribe-Baumann@eah-jena.de

³ Ernst-Abbe-Hochschule Jena, Grundlagenwissenschaften, Carl-Zeiss-Promenade 2, 07745, Jena, Christina.Class@eah-jena.de

⁴ Universität Osnabrück, Institut für Informatik, Albrechtstr. 28, 49076, Osnabrück, Michael.Brinkmeier@uni-osnabrueck.de

Thinking-Fähigkeiten unterstützen, indem das automatisierte Lösen von Problemen geschult wird [Wo17]. Die Einführung in eine Programmiersprache findet häufig bereits zu Beginn einer Informatikausbildung statt.

Um genauer beleuchten zu können, wie Fachkonzepte der Programmierung so erschlossen werden können, dass sie letztendlich zur Lösung von Problemen eingesetzt werden können, betrachten wir die kognitive Psychologie des Lösens von Problemen. In diesem Kontext ist unser Ziel die Verankerung von Fachkonzepten in Wissensstrukturen, sodass sie als Operatoren im Sinne der Problemlösetheorie eingesetzt werden können. Hierbei gehen wir von folgender Begriffsbildung aus [Sp03]:

- *Problem als Zustand*: Ein Problem kann aufgefasst werden als ein Ausgangszustand. Die Lösung des Problems entspricht einem Zielzustand.
- *Lösungsprozess als Transformation*: Um den Ausgangszustand in den Zielzustand zu transformieren, müssen Problemlöseoperatoren angewendet werden. Eine Transformation besteht aus einer konkreten Reihenfolge von Operationen unter Einsatz geeigneter Operatoren und kann mehrere Übergangszustände beinhalten.

Wir können diese Begriffe auf die Programmierung übertragen: Ein zu lösendes Ausgangsproblem kann durch eine geeignete Repräsentation [An13] als *Ausgangszustand* eines Programmes betrachtet werden. Die Durchführung von Operationen, die durch das Programm festgelegt sind, stellt den *Lösungsprozess* dar. Der Zustand des Programms wird während seiner Durchführung transformiert, bis ein definierter Endzustand erreicht ist. Der Endzustand ist eine Repräsentation der Lösung des Problems. Um geeignete Operationen für den Lösungsprozess zu finden, müssen entsprechende Fachkonzepte aus der Programmierung als Problemlöseoperatoren eingesetzt werden.

Ein zentrales Problem in der Ausbildung von Programmieranfängern ist also der Erwerb von entsprechenden Problemlöseoperatoren. Nach der Theorie des Problemlösens gemäß Anderson [An13] erfolgt dieser Erwerb durch Entdecken, direkte Instruktion oder den Einsatz von Analogien. Im Abschnitt 2 werden Probleme diskutiert, die bei der allgemeinen Vermittlung von Programmierkonzepten auftreten. Danach richten wir den Fokus auf den Einsatz von Analogien, denen eine möglichst große Bildhaftigkeit innewohnt. Im Abschnitt 3 werden unsere Erwartungen an bildhafte Analogien und die Voraussetzungen für deren Verwendung erläutert. Anschließend werden im Abschnitt 4 spezifische Aspekte, die bei der Verwendung von bildhaften Analogien auftreten können, anhand von Beispielen geschildert.

Die Inhalte dieses Beitrags sollen in einem Workshop vertieft werden. Dazu werden in Arbeitsgruppen Praxisbeispiele von Analogien für konkrete Programmierkonzepte zusammengestellt. Deren Vor- und Nachteile sowie Grenzen und resultierende Herausforderungen werden anschließend diskutiert.

2 Herausforderungen bei der Vermittlung von Programmierkonzepten

Trotz der Tatsache, dass einige Lernende auch ohne Vorkenntnisse zum Bereich der Programmierung einen leichten Zugang finden und die vermittelten Konzepten schnell einsetzen können, stellen Inhalte, die bereits am Anfang der Programmierausbildung stehen (z.B. Variablen, Datentypen, Verzweigung, Iteration), für einen erheblichen Teil der Studierenden oft schwer überwindbare Hürden dar.

Diese Beobachtung wird in Abbildung 1 illustriert durch die Auswertung von Prüfungen des ersten Semesters im Bereich Informatik von Studierenden einer Ingenieurwissenschaft. Während man möglicherweise andere Verteilungen der Noten erwarten würde, tritt ein in der landläufigen Meinung von Lehrenden als typisch angesehenes Phänomen auf: Es gibt eine Häufung von guten und sehr guten ebenso wie von schlechten und sehr schlechten Noten bei vergleichsweise wenigen mittelmäßigen Noten. Nach Meinung der Autoren deutet diese Notenverteilung auf eine große Anzahl an Studierenden hin, die keinerlei oder fast keinen Zugang zu den entsprechenden Konzepten finden konnten. Insbesondere gibt es recht wenige Studierende, deren Leistungen im Mittelfeld liegen und die recht passabel mit dem Stoff und damit den Konzepten zurechtkommen.

Wir führen dies darauf zurück, dass Fachkonzepten aus dem Bereich der Programmierung oft ein hoher Grad an Abstraktion innewohnt. Um Konzepte erfolgreich einsetzen zu können, müssen in hohem Ausmaß formale Denkopoperationen durchgeführt werden. Bereits der Entwurf einfacher Programme geht oft mit einem Prozess der Hypothesenbildung einher, zum Beispiel um Nutzereingaben, die vor Ausführung des Programmes unbekannt sind, zu verarbeiten oder Iterationen, deren Häufigkeit vom Zustand des Programmes abhängt, zu realisieren. Auch die Verwendung von

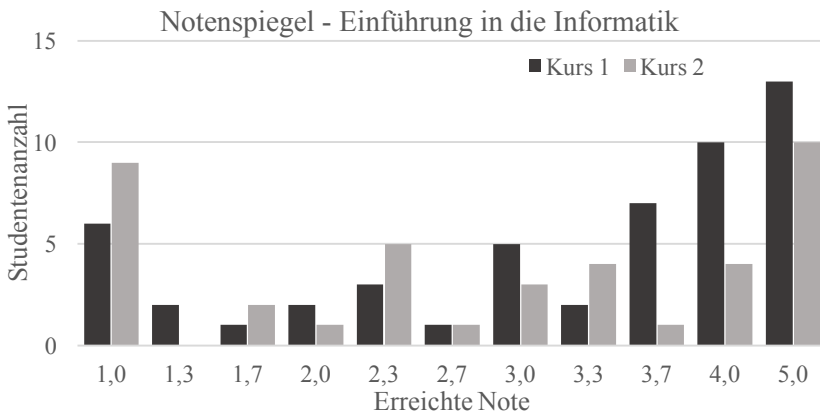


Abbildung 1: Notenspiegel von zwei Informatik-Prüfungen im ersten Studiensemester

Hilfsvariablen, zum Beispiel für die Bildung einer Summe über die Elemente einer Liste, geht über das konkret-operative Denken hinaus, auf welches der Lernende hinsichtlich des Problemlöseprozesses zurückgreifen kann.

Hinsichtlich der Hilfestellung für Lernende mit Schwierigkeiten haben die Autoren wenig Erfolg mit schrittweisem, ausführlichem Erläutern von einfachen Quelltextbeispielen beobachtet. Wesentlich bessere Erfahrungen wurden hingegen mit bildhaften Vergleichen gemacht, um betroffene Konzepte anschaulich zu machen. Bei der Veranschaulichung wird entweder Bezug auf Objekte aus der Lebenswelt der Lernenden genommen oder auf Prozesse, die aufgrund der Erfahrung der Lernenden bekannt oder leicht zugänglich sind. Somit werden Verstehensprozesse hinsichtlich deklarativen oder prozeduralen Wissens unterstützt.

3 Erwartungen und Voraussetzungen an die Verwendung bildhafter Analogien

Schwierigkeiten beim Zugang zu Fachkonzepten sollen also durch den Einsatz von bildhaften Analogien gemindert werden. Die Nutzung von bildhaften Analogien hat folgende Vorteile:

- Ist eine Analogie ausreichend bildhaft und aus dem Erfahrungsbereich des Lernenden gegriffen, so kann eine Einbindung an die Wissensstrukturen des Lernenden erleichtert werden. Dies stellt wichtige Dimensionen im verständnisintensiven Lernen dar [ViL17]. Betrachten wir das Vorwissen des Lernalters und das zu vermittelnde Fachwissen als Objekte, so kann mit einer bildhaften Analogie eine Verknüpfung zwischen Vorwissen und Fachwissen wie in Abbildung 2 hergestellt werden.
- Die Anknüpfung an vorhandenes Wissen kann beim Lernenden bereits zu einer positiven Selbsteinschätzung hinsichtlich des Lernprozesses führen und damit zu einer Erhöhung der Lernmotivation [DR13].
- Im fortgeschrittenen Lernprozess, nachdem das Fachwissen und das Verständnis für die Analogie sich gefestigt haben, kann eine duale Kodierung des Wissens erreicht werden: Das erworbene verbale Wissen wird durch die nonverbale Bildhaftigkeit der Analogie ergänzt. Es ist anzunehmen, dass dual kodiertes Wissen besser abgerufen werden kann [Ma01] und dadurch ebenso besser auf andere Situation übertragen und somit als Problemoperator eingesetzt werden kann.
- In umgekehrter Richtung kann ein Fachkonzept Einzug in alltägliche Situationen

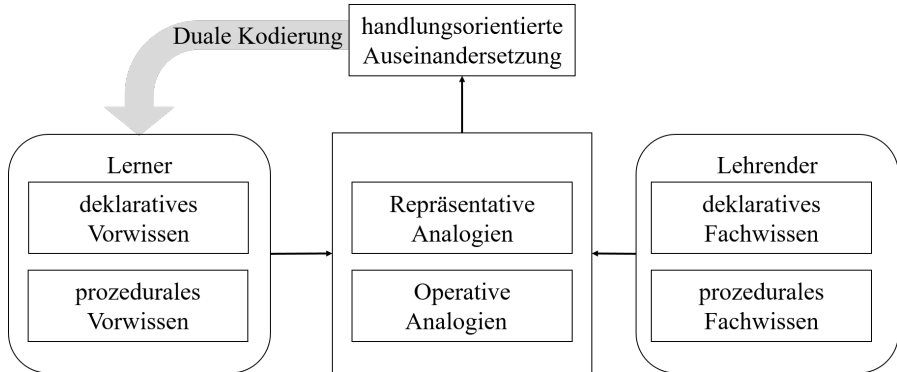


Abbildung 2: Vorwissen und Fachwissen mit Analogien zusammenführen und mithilfe handlungsorientierten Auseinandersetzungen zum dual-kodierten Wissen des Lerners führen.

erhalten, wenn ein Objekt als Analogon verwendet wird, welchem ein Lernender häufig begegnet. Somit kann eine Veränderung der Betrachtungsweise alltäglicher Gegenstände und Prozeduren hinsichtlich des Computational Thinking möglich sein. Wird z.B. ein Snackautomat als Analogon für eine Funktion verwendet (siehe Abschnitt 4.4), erinnert sich der Lernende ggf. bei künftigen Snackautomaten-Einkäufen an den Funktionsbegriff.

Analogien werden in der Lehre und in Lehrbüchern schon lange eingesetzt, und bereits im letzten Jahrtausend gab es umfangreiche Untersuchungen zu deren Einsatz in den Naturwissenschaften. In einer Literaturstudie stellte Duit die Ergebnisse verschiedenster Untersuchungen zusammen und wies auf diverse Probleme hin [Du91]. Bereits 1982 wurde vor dem unbedachten Einsatz von Analogien in der Informatikbildung gewarnt, da diese das Lernen erschweren könnten [HM82].

Voraussetzung für den Einsatz einer Analogie ist daher eine hohe Passgenauigkeit. Hierzu müssen die Elemente eines Fachkonzeptes, die durch den Einsatz einer Analogie abgedeckt werden, identifiziert werden. Um die erfolgreiche Verknüpfung der Analogie zum Fachkonzept herzustellen, ist laut Anderson sicherzustellen, dass bei „jedem Akt des Analogiebildens“ die „Elemente von der Quelle auf das Ziel [...] übertragen“ werden [An13]. Eine bloße Nennung eines Bildes im Umfeld der Vorstellung eines Fachkonzeptes ist demzufolge nicht ausreichend. Studien haben gezeigt, dass oft eine Anleitung notwendig ist, um die Analogien richtig in die Denkprozesse zu integrieren [Du91]. Auch ist wichtig, auf die Unterschiede zwischen Analogie und der Quelle hinzuweisen, um deutlich zu machen, wo die Gemeinsamkeiten nicht helfen. Hier kann es hilfreich sein, verschiedene Analogien für die einzelnen Aspekte zu verwenden [Du91].

Das Verständnis von Konzepten ist im Bereich der Programmierung oftmals nicht ausreichend. Weinberg [We98] stellte in seiner Beobachtung von Lernen von Programmiersprachen fest, dass ein Wissenserwerb über Fachkonzepte möglicherweise nicht stattfinden kann ohne handlungsorientierte Auseinandersetzung mit dem

entsprechenden System.

Entscheidend ist also die Verzahnung der Wissensvermittlung durch den Lehrenden und des eigenständigen Handelns durch den Lernenden. Hierbei kann die Verwendung von Analogien zum Einstieg in ein Programmierkonzept hilfreich sein, ein handelnder Bezug zum Programmieren muss dennoch hergestellt werden. Für den Wissenserwerb soll ein Lernender jedes neue Konzept aktiv anwenden und entsprechende Programme am Rechner implementieren. Der Bezug zur Analogie kann während dieses Prozesses immer wieder unterstützend hergestellt werden. Dies ist jedoch nur so lange sinnvoll, wie die relevanten Gemeinsamkeiten der Analogien zum Tragen kommen und Lernende nicht verwirrt werden. Aus diesem Grund kommen Auswahl und zielführendem Einsatz von Analogien eine große Bedeutung zu. Hierbei gilt es sowohl die Frage nach der Wahl der Analogien zu betrachten, als auch zu entscheiden, wann von der Verwendung der Analogie abgesehen werden und der Lernende auf relevante Unterschiede hingewiesen werden soll.

4 Probleme und Fragen hinsichtlich des Einsatzes von bildhaften Analogien

Um zu verdeutlichen, wie ein Programm einen Problemlöseprozess im Sinne von Anderson darstellt, soll das folgende Beispiel betrachtet werden. Es soll für eine Menge von Zahlen in einem Programm das Minimum dieser Zahlen bestimmt werden. Eine Automatisierung des Lösungsprozesses in der Programmiersprache Python könnte folgendermaßen aussehen:

```
liste = [2,4,6,3,5]
minimum = liste[0]
zaehler = 1
while zaehler<len(liste):
    if minimum > liste[zaehler]:
        minimum = liste[zaehler]
    zaehler = zaehler + 1
print(minimum)
```

Die Repräsentation des Problems beinhaltet die Variablen *liste*, *zaehler* und *minimum* mit ihren anfänglich definierten Werten. Der Zielzustand ist erreicht, wenn die Variable *minimum* das tatsächliche Minimum der Zahlen bezeichnet. Um durch geeignete Operationen zum Zielzustand zu gelangen, wird eine Reihe von Transformationen angewendet:

- *zaehler* wird in jeder Iteration inkrementiert.
- Ein Vergleich von aktuellem Listenelement und aktuellem Minimum findet statt, falls nötig, wird *minimum* angepasst.

- Ein Abbruchkriterium wird überprüft.

Notwendige Konzepte für die Operationen sind hierbei die Wertzuweisung, die Auswertung logischer Ausdrücke sowie die Iteration.

4.1 Funktionale Fixierungen

Unter funktionaler Fixierung wird verstanden, dass für einen Problemlöseoperator nicht der volle Umfang seiner Möglichkeiten erkannt wird, sondern er nur in ausgewählten (bekannten) Kontexten eingesetzt wird. Der von Anderson [An12] beschriebene Effekt der funktionalen Fixierung im Zusammenhang mit Problemlöseoperatoren hat womöglich auch Bedeutung für die Verwendung von bildhaften Analogien.

Während Programmierkonzepte allgemeingültigen Charakter haben, sind bildhafte Analogien häufig in einer konkreten Situation verankert. Für jede bildhafte Analogie muss untersucht werden, inwieweit die Gefahr der Einschränkung des Operators im Sinne einer funktionalen Fixierung durch fehlende allgemeingültigen Charakter der Analogie besteht.

Mit der Vielfalt an Ausprägungen von Programmierkonzepten stellen sich daher beim didaktischen Einsatz von Analogien immer die Fragen:

- Wie lange kann diese als übergangsweise Veranschaulichung dienen, um später zu einer Abgrenzung zu gelangen?
- Wie sehr ist eine Vereinfachung eines entsprechenden Programmierkonzeptes möglich und wie wird diese später korrigiert?

4.2 Deklarative Analogie: Schachtel als Analogie für Variablenkonzept

Unter einer *deklarativen Analogie* verstehen wir eine Analogie, welche Verstehensprozesse unterstützt, die auf deklaratives Wissen bezogen sind. Häufig wird z.B. das Bild eines beschrifteten Gefäßes (oder Schachtel) verwendet, um das Konzept einer Variablen einzuführen. In obigem Beispiel könnte man mit einer Schachtel darstellen, welche Rolle *minimum* im Algorithmus hat.

Aber auch wenn hierfür die von Anderson geforderten Passgenauigkeit einer Analogie erfüllt ist, repräsentiert das Bild der Schachtel das Konzept der Variable als Ganzes noch nicht völlig korrekt:

- In der Realität kann eine Schachtel leer sein. Dies trifft auf Variable nicht zu, da sie als Speicherbereich immer einen, wenn auch zufälligen, Wert enthalten. In Sprachen wie C, welche das Lesen einer Variable ohne vorherige Initialisierung ermöglichen, ist dies ein relevanter Unterschied. Dies kann also zu einer falschen Vorstellung einer Variablenzuweisung führen.

- In der Realität kann eine Schachtel durchaus mehrere Elemente enthalten. Eine Zuweisung eines Wertes an eine Variable führt aber dazu, dass der vorherige Wert unwiderruflich überschrieben wird.
- In der Realität hat eine Schachtel eine gegebene Größe. In C/C++ wird zwar durch Deklaration einer Variablen ein bestimmter Speicherbereich bezeichnet, in welchem Daten des deklarierten Typs gespeichert sind. In Python kann das Bild der Schachtel aber zu einer fehlerhaften funktionalen Fixierung führen, da die gleiche Variable im Laufe der Zeit Werte unterschiedlichen Typs und unterschiedlicher Größe speichern kann. Eine Variable in Python stellt lediglich den Namen eines Objektes im Speicher dar - durch Zuweisung eines anderen Wertes bezeichnet die Variable meist ein anderes Objekt im Speicher.

Hat sich die Vorstellung einer Schachtel als Analogon zu einer Variable manifestiert, können Probleme auftreten, wenn weitere Konzepte eingeführt werden, z.B. statische und dynamische Variablen oder Listen. Diese sind mit dem Schachtelmodell nicht unmittelbar zugänglich und bei Einführung dieser muss Abstand von der bisher verwendeten Analogie genommen werden oder weitere Analogien hinzugefügt werden, um die anderen Eigenschaften aufzuzeigen. Solche multiplen Analogien können notwendig sein, um einen Konzept umfassend zu erlernen [Du91].

4.3 Deklarative Analogie: Snackautomat als Analogie für Funktionskonzept

Die Kapselung von Quelltext in Funktionen ist ein wichtiges Konzept in der Programmierung. Funktionen stellen bereits ein relativ hohes Abstraktionsniveau dar und bereiten Lernenden häufig Schwierigkeiten. Ein grundlegendes Problem ist die Trennung von *Definition* und *Ausführung* von Funktionen. Lernenden fällt es oft schwer zu verstehen, dass Funktionen erst durch einen Funktionsaufruf ausgeführt werden und Parameter erst durch den Funktionsaufruf konkrete Werte erhalten.

Als bildhafte Analogie für eine Funktion mit Rückgabewert, unter Verwendung zweier Parameter, kann ein Snackautomat dienen. Ein Funktionsparameter steht für den Geldbetrag, der in den Automaten eingeworfen wird, ein weiterer für die Nummer, die zur Auswahl des Produktes über eine Tastatur eingegeben wird. Während der Funktionsabarbeitung werden verschiedene Operationen von dem Snackautomaten durchgeführt und als Rückgabewert erhält man das gewünschte Produkt. Das Ausgabefach kann als Analogie für das *return*-Statement in der Funktionsdefinition betrachtet werden.

Das Beispiel knüpft an die Vorstellungswelt der Lernenden an und ist hilfreich, um das Konzept von Funktion, Funktionsaufruf und Aufrufer zu verdeutlichen. Dennoch kann auch hier eine funktionale Fixierung erfolgen, wenn der Lernende zu einer der folgenden Schlussfolgerungen gelangt:

- Eine Funktion hat immer zwei Parameter.
- Eine Funktion muss immer einen Rückgabewert liefern.
- Der Rückgabewert einer Funktion muss immer von allen Parametern abhängen.

Um eine Manifestation dieser Fixierung zu verhindern, müssen gezielt Verallgemeinerungen anhand geeigneter Programmieraufgaben stattfinden, z.B. von:

- Einer Funktion mit nur einem Argument.
- Einer Funktion ohne Parameter.
- Einer Funktion ohne Rückgabewert.
- Einer Funktion, deren Rückgabewert nur von einer Teilmenge ihrer Parameter abhängt.

4.4 Operative Analogie: Finden der kürzesten Spielzeit aus einer CD-Regal als Analogie für Bestimmung des Minimums

Unter einer *operativen Analogie* verstehen wir eine Analogie, welche Verstehensprozesse unterstützt, die auf prozedurales Wissen bezogen sind. Das Abarbeiten eines Rezepts kann z.B. als operative Analogie für das Abarbeiten eines Algorithmus dienen.

Mit Hilfe einer operativen Analogie kann der Prozess zur Entwicklung des obigen Algorithmus zur Minimumbestimmung unterstützt werden. In diesem Zusammenhang wird folgende analoge Aufgabe gestellt: Die kürzeste Spieldauer von CDs in einem Regal soll gefunden werden. Durch eigene Überlegungen kommen Lernende schnell auf die Idee, dass es sinnvoll ist, alle CDs der Reihe nach zu betrachten und sich die bisherige kürzeste Spieldauer zu merken. Wenn das Problem des Minimums mehrerer Zahlen gestellt wird, muss der Lernende Gemeinsamkeiten mit der Analogie identifizieren, um einen Transfer des Lösungsprozesses vorzunehmen: Das Regal ist eine Liste; die CDs sind die Zahlen; die gemerkte kürzeste Spielzeit ist eine Variable. Durch die Analogie werden die Lernenden dabei unterstützt, eigene Algorithmen mithilfe ihrem prozeduralen Vorwissen zu entwickeln.

5 Zusammenfassung

Die Nutzung von bildhaften Analogien für Fachkonzepte, die im Rahmen der Programmierausbildung vermittelt werden sollen, dient als Hilfsmittel, um anfängliche Verständnisschwierigkeiten zu überwinden. Wichtig hierbei ist eine wiederholte Auseinandersetzung mit der Analogie während der Bearbeitung von Programmieraufgaben, welche im Zusammenhang mit dem zu lernenden Fachkonzept

stehen, um Fehlvorstellungen durch mangelnde Passgenauigkeit und Effekte funktionaler Fixierung zu vermeiden. Analogien sollen helfen, Programmierkonzepte besser in Wissensstrukturen zu verankern, um Problemlösefähigkeiten in Bezug auf Programmierung zu verbessern. Ebenso sollen Analogien es ermöglichen, vermehrt alltägliche Situationen aus dem Blickwinkel des Computational Thinking zu betrachten.

Literaturverzeichnis

- [An13] Anderson, J.R.: Kognitive Psychologie. VS Verlag für Sozialwissenschaften, S. 165, 249, 2013.
- [BS11] Barr, V; Stephenson, C.: Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community?. *acm Inroads*, 2/1, S. 48, 2011.
- [De09] Denning, P.J.: The Profession of IT Beyond Computational Thinking, *Communications of the ACM*, 52/6, S. 28-30, 2009.
- [DR93] Deci, E.L.; Ryan, R.M.: Die Selbstbestimmungstheorie der Motivation und ihre Bedeutung fuer die Paedagogik. *Zeitschrift für Pädagogik*, 39/2, S. 223-238, 1993.
- [Du91] Duit, R.: On the Role of Analogies and Metaphors in Learning Science, *Science Education*, 75/6, S. 649-672, 1991.
- [HM82] Halasz, F.; Moran, T.: Analogy Considered Harmful, *CHI '82 Proceedings of the 1982 Conference on Human Factors in Computing Systems*, S. 383-386, 1982.
- [Ma01] Martschinke, S.: Aufbau mentaler Modelle durch bildliche Darstellungen. Waxmann Verlag, Münster [u.a.], S. 61, 2001.
- [Sp03] Spring, M.: "Praktisches Problemlösen", 2003, http://www.psychologie.uni-heidelberg.de/ae/allg/mitarb/ms/PH_Einfuehrung.pdf, zuletzt besucht am 9.6.2017.
- [ViL17] Verein für Verständnisintensives Lernen e.V., http://www.verstehenlernen.de/?page_id=672, zuletzt besucht am 6.6.2017.
- [We98] Weinberg, G.M.: *The Psychology of Computer Programming*. Silver Anniversary ed, Dorset House Publ., 1998.
- [Wi06] Wing, J.: Computational Thinking, *Communications of the ACM*, 49/3, S. 33-35, 2006.
- [Wo17] Stephan Wolfram Blog, <http://blog.stephenwolfram.com/2016/09/how-to-teach-computational-thinking/>, zuletzt besucht am 9.3.2017.
- [YSH17] Yadav, A; Stephenson C.; Hong, H.: Computational Thinking for Teacher Education, *Communications of the ACM*, 60/4, S. 55-62, 2017.

Workshop: Datenbank-Anfragen mit SQL und dem elektronischen Schulbuch inf-schule.de

Heiko Jochum¹, Klaus Becker², Martin Zimnol³, Daniel Jonietz⁴, Oliver Schneider⁵,
Manuel Froitzheim⁶

Abstract: Anhand einer konkreten Unterrichtseinheit zum Thema Datenbanken wird exemplarisch die Arbeit mit dem elektronischen Schulbuch inf-schule.de vorgestellt. Die Unterrichtseinheit beruht auf der Verwendung einer auch online verfügbaren Datenbank und wird anhand vieler Übungsphasen nachvollzogen. Die Teilnehmenden benötigen nur ein Endgerät mit Internetzugang. Für die Datenbankabfragen kann alternativ auch ein bereits installiertes XAMPP-Paket verwendet werden.

Keywords: Elektronisches Schulbuch, Digitales Schulbuch, Informatikunterricht, Lernen, Didaktik der Informatik, Web-Applikation, Datenbank

1 Einleitung und Motivation

Digitale Bildung wird inzwischen als unverzichtbar in einer immer stärker digital geprägten und vernetzten Welt angesehen. Die Aneignung grundlegender Konzepte zum Verständnis dieser Welt und die Entwicklung von Kompetenzen zur Mitgestaltung einer solchen Welt sollte in einem eigenständigen Lernbereich erfolgen – so die einhellige Forderung aller mit digitaler Bildung befassten Fachdidaktiker [Da16]. Nur in einem fundierten Informatikunterricht kann die anwendungsbezogene Perspektive (Wie nutze ich das?) und die gesellschaftlich-kulturelle Perspektive (Wie wirkt das?) um die zum Verständnis der digitalen Welt erforderliche technologische Perspektive (Wie funktioniert das?) erweitert werden ([F+17], ebenfalls in diesem Band).

Der Erwerb digitaler Bildung wird im Informatikunterricht gezielt gefördert. Das Schulbuch als Leitmedium bei der Unterrichtsplanung und Unterrichtsgestaltung spielt dabei eine zentrale Rolle [Wi03], weil hier „Lerninhalte eines Faches oder Lernbereichs in systematischer, didaktischer und methodischer Form“ bereitgestellt werden [Sa10].

¹ Werner-Heisenberg-Gymnasium Bad Dürkheim, Kanalstr. 19, 67098 Bad Dürkheim, jom@whgonline.de

² BurgGymnasium Kaiserslautern, Burgstraße 18, 67659 Kaiserslautern, beckerkp@web.de

³ Pädagogisches Landesinstitut Rheinland-Pfalz, Butenschönstraße 2, 67346 Speyer, martin.zimnol@pl.rlp.de

⁴ BurgGymnasium Kaiserslautern, Burgstraße 18, 67659 Kaiserslautern, daniel@jonietz.de

⁵ Sickingen-Gymnasium Landstuhl, Philipp-Fauth-Straße 3, 66849 Landstuhl, o.schneider@mac.com

⁶ Universität Siegen, Zentrum für ökonomische Bildung in Siegen (ZöBiS), Kohlbettstraße 15, 57068 Siegen, froitzheim@zoebis.de

Schulbücher wurden bisher in Buchform erstellt. Mit der Entwicklung digitaler bzw. elektronischer Schulbücher ist ein neues Forschungsfeld entstanden, das der Entwicklung von Lernumgebungen neue Impulse verleiht [ScFr14][ScFr15][ScFr16].

2 Workshopbeschreibung

Innerhalb des Workshops wird die Arbeit mit dem Schulbuch inf-schule.de anhand einer Unterrichtseinheit zu dem Thema „Datenbanksysteme“ vorgestellt. Ziel ist es, die Unterrichtseinheit durch eigenständige Arbeit am Computer nachzuvollziehen.

Die Unterrichtseinheit [Mo16] beruht auf der sukzessiven Normalisierung und Erweiterung einer vorhandenen Datenbank für Geodaten. Durch die Verwendung einer vorhandenen Datenbank müssen die Schülerinnen und Schüler keinen Datenbankentwurf vornehmen, bevor sie nicht die Arbeit mit einer Datenbank kennengelernt haben. Durch verschiedene Versionen erkennen die Schülerinnen und Schüler zum einen die Notwendigkeit der Vermeidung von Redundanz und können sich zum anderen durch Verfolgung des didaktischen Prinzips der Isolation von Schwierigkeiten zunächst auf wenige Tabellen (z.B. Länder, Orte) innerhalb der Datenbank konzentrieren, bevor diese nach und nach durch weitere Tabellen (z.B. Flüsse) ergänzt werden.

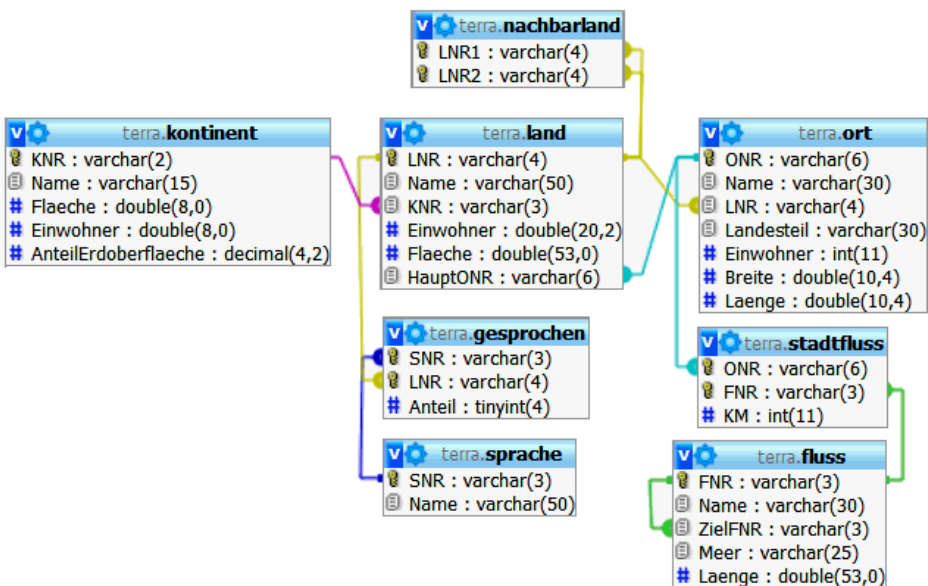


Abbildung 1: Datenbankschema der vollständigen Geodatenbank "terra" (vgl.: <https://inf-schule.de/information/datenbanksysteme/terra/selbstreferenz/uebungen>)

Die Durchführung der SQL-Anfragen an die Datenbank kann zum einen durch online-Anfragen innerhalb des Schulbuchs inf-schule.de erfolgen, zum anderen ist es auch möglich, die verschiedenen Datenbank-Versionen auf einem eigenen Server bzw. einem lokalen Computer zu installieren und auf diesem Wege die Anfragen zu stellen.

Literaturverzeichnis

- [Da16] Dagstuhl-Erklärung: Bildung in der digitalen vernetzten Welt, www.gi.de/fileadmin/redaktion/Themen/dagstuhl-erklaerung-bildung-in-der-digitalen-welt-2016.pdf, Stand: 23.01.2017.
- [F+17] Froitzheim, M.; Becker, K.; Zimmol, M.; Jonietz, D.; Schneider, O.; Jochum, Heiko: Förderung digitaler Bildung mit digitalen Schulbüchern. (ebenfalls in diesem Band)
- [Mo16] Mohr, T.: Datenbanksysteme, www.inf-schule.de/information/datenbanksysteme, Stand: 10.02.2017.
- [Sa12] Sanguo, C.; Xuehai, M.; Schenglin, L. (2012). The Pace of Ebook Development in China. IN: LOGOS: The Journal of the World Book Community, 23(2), 14-20.
- [ScFr14] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [ScFr15] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch 2015. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [ScFr16] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch 2016. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [Wi03] Wiater, W. (2003): Schulbuchforschung in Europa. Bestandsaufnahme und Zukunftsperspektive. Bad Heilbrunn: Klinkhardt.

Online kooperieren im Informatikunterricht

Thomas Rohde¹, Kolja Strauss², Gregor Alexander Benedikt³, René Levens⁴, Lina Haji⁵,
Olaf Müller⁶

Abstract: Der Verzicht auf kooperative Lernformen in Rechnerräumen mit der Begründung, dass diese aufgrund der meist unveränderlichen Tischanordnung und ortsfest montierten Rechner nur schwer zu realisieren seien, ist üblich – wird aber der Forderung nach kompetenzorientiertem Unterricht nicht gerecht. Die Begründung ist dennoch nachvollziehbar, da es an Konzepten mangelt, die aus der räumlich und technisch bedingten Not eine Tugend machen. Daher wird hier anhand der bekannten Methoden „Placemat“, „Kartenabfrage“, „Evaluationszielscheibe“ und „Lerntempoduett“ beispielhaft gezeigt, wie das Potenzial der im Raum befindlichen Rechner genutzt werden kann, um auf Basis von Webanwendungen Kooperation zu ermöglichen. Darüber hinaus wird ein „Methodenkoffer“ vorgestellt, mit dem sich die Onlinekooperation unter Einsatz von schülereigenen Smartphones/Tablets in jedem Klassenraum realisieren lässt, der über einen WLAN-fähigen PC mit Beamer verfügt. Die vorgestellten Lernwerkzeuge bieten weitere Vorteile wie hohe Flexibilität im Einsatz, Unabhängigkeit von kostspieligen und umweltschädlichen Materialien (insb. Papier) sowie Möglichkeiten einer unkomplizierten dauerhaften Sicherung und Wiederverwendbarkeit der Arbeitsergebnisse.

Keywords: Kooperative Lernform, Think-Pair-Share, Informatikunterricht, Webanwendung, BYOD, Raspberry Pi, Placemat, Lerntempoduett, Kartenabfrage, Metaplankarten, Brainstorming, Evaluationszielscheibe.

1 Einleitung

Die Realisierung von Methodenvielfalt im Informatikunterricht ist eine besondere Herausforderung, da Tische und Hardware im Rechnerraum meist ortsfest aufgestellt sind und ohne unverhältnismäßig großen Aufwand nicht verändert werden können. Häufig stehen die Tische zusätzlich in engen Reihen oder U-Form, so dass gemeinsames Arbeiten um einen Gruppentisch herum unmöglich wird. Kleine Tafeln oder Whiteboards limitieren die Menge der anschreibbaren Informationen und für gewöhnlich fehlen Pinnwände oder Magnetleisten an den Wänden. Weiterhin zwingt der Nachhaltigkeitsgedanke zum schonenden Umgang mit Ressourcen wie Papier, Stiften, Klebstoff usw. Andererseits verfügen die Schülerinnen und Schüler (SuS) über vernetzte

¹ BBS Lingen – Agrar und Soziales, Beckstraße 23, 49809 Lingen, rohde@bbs-lingen-as.net

² GBS Grafschaft Bentheim, Denekamper Str. 1, 48529 Nordhorn, strauss@gbs-grafschaft.de

³ Berufskolleg Lübbecke, Rahdener Str. 1, 32312 Lübbecke, benedikt@berufskolleg-lk.de

⁴ BBS Gesundheit und Soziales, Am Bölt 5, 48527 Nordhorn, rene.levens@bbs-gs-nordhorn.de

⁵ BBS Brinkstraße, Brinkstraße 17, 49080 Osnabrück, haji@bbs-os-brinkstr.de

⁶ Studienseminar für das Lehramt an BBS Osnabrück, Blumenthalstraße 32, 49076 Osnabrück, o.mueller@studienseminar-os.de

und universell einsetzbare Werkzeuge in Form von Standrechnern oder Smartphones bzw. Tablets. Diese Konstellation ist vor dem Hintergrund der in (niedersächsischen) Curricula geforderten Förderung der Sozialkompetenz inakzeptabel.

Werkzeuge zur Online-Kollaboration gibt es bereits einige, siehe z.B. [St14]. Leider sind sie entweder kostenpflichtig⁷ und/oder sie unterstützen die gewünschte Phasierung (z.B. Think-Pair-Share) des Unterrichts nicht. Daher wurden im Fachseminar Informatik am Studienseminar für das Lehramt an Berufsbildenden Schulen in Osnabrück einige bekannte Unterrichtsmethoden in Form von Webanwendungen in Online-Versionen überführt.

Alle Online-Versionen der Methoden sind zum direkten Einsatz im Unterricht unter <http://www.oncoo.de> erreichbar.

2 Die Methoden

Im Folgenden werden die bisher als Webanwendung realisierten Methoden jeweils unter Nennung der Nachteile der klassischen offline bzw. tafel-/papierbasierten Variante sowie Skizzierung der Vorteile der Online-Version dargestellt. Alle Webanwendungen sind in Standardbrowsern lauffähig. Die Anforderungen unterscheiden sich nur geringfügig und der Einsatz kann z.T. in verschiedenen Unterrichtsphasen erfolgen (s. Tab. 1).

Methode	Lehrer-Client	SuS-Clients	Java-Script im SuS-Client	Unterrichtsphase
Placemat	PC	PC, Tablet	ja	Erarbeitung
Evaluationszielscheibe	PC	PC, Tablet, Smartphone	nein	Reflexion
Kartenabfrage	PC	PC, Tablet, Smartphone	ja	Planung, Erarbeitung
Lerntempoduett	PC	PC, Tablet, Smartphone	ja	Erarbeitung

Tab. 1: Anforderungen und Einsatzmöglichkeiten der einzelnen Methoden

2.1 Das Placemat

Einsatzzweck, Ablauf und Vorteile der Placemat-Methode wurden z.B. in [Re14] oder [GG06] bereits ausführlich beschrieben. Die Nachteile der klassischen Variante der Methode auf Papier sind:

- Es wird pro Gruppe ein Tisch benötigt, um den die Mitglieder gemeinsam sitzen können.

⁷ Es wird z.B. entweder Geld oder die Herausgabe personenbezogener Daten (z.B. Telefonnummer) verlangt.

- Rechteckige Blätter lassen sich bei drei Gruppenmitgliedern nicht in drei gleichgroße Bereiche teilen.
- Während der Think-Phase ist „Spicken“ möglich.
- Das sukzessive Lesen/Betrachten/Erfassen der Einzellösungen zu Beginn der Pair-Phase gelingt den einzelnen Schülerinnen und Schülern unterschiedlich schnell. Einige müssen warten, bis das Placemat weiter gedreht werden kann.
- Während der Diskussion in der Pair-Phase ist der Großteil der Teillösungen für jedes Gruppenmitglied verdreht.
- Am Ende der Pair-Phase ist das Gruppenergebnis für alle SuS verdreht, die gerade nicht daran schreiben.
- Das Gruppenergebnis muss von einem Gruppenmitglied mit leserlicher Handschrift komplett neu aufgeschrieben werden, obwohl es zum größten Teil aus Formulierungen der Teilergebnisse besteht.
- Die Präsentation des Gruppenergebnisses durch Aufhängen des Plakates in der Share-Phase ist in manchen Klassen- und vielen Fachräumen problematisch.
- Das Plakat kann in Fachräumen häufig nicht bis zur nächsten Stunde verbleiben.

Die Webanwendung Online-Placemat wurde entwickelt, um die Methode Placemat im Rechnerraum unter Umgehung der genannten räumlichen Probleme und Verbesserung der Nachteile der Papierversion zu realisieren.


	Think-Phase endet in: 0:00:49		Pair-Phase endet in: 0:10:49	
	A	B	C	D
	Sarah	Anna	Chiara	Tristan
	Michael B.	Anastasia	Julia	Niklas
	Leutrim	Franzsika	Michael K.	Cao
	Lisa ✉	Lars ✉	Florian ✉	
	Jessica	Jan	Fatih ✉	

Abb. 1: Dauerhaft über den Beamer projizierte Gesamtansicht mit Informationen zu den verbleibenden Zeiten, Gruppenzusammensetzungen und Bearbeitungsständen

Die Lehrkraft legt auf ihrer Startseite der Webanwendung zunächst die Gruppengrößen sowie die Dauer der Think- und die der Pair-Phase fest. Dann erscheint die

Gesamtansicht, die dauerhaft in der Projektion sichtbar bleibt und es erlaubt, sowohl die Gruppenzuordnung als auch den Bearbeitungsstand der SuS zu überblicken (s. Abb. 1: Dauerhaft über den Beamer projizierte Gesamtansicht mit Informationen zu den verbleibenden Zeiten, Gruppenzusammensetzungen und Bearbeitungsständen).

Die einzelnen SuS registrieren sich auf der Schülerstartseite mit ihrem Namen und werden per Zufall einer Farbgruppe zugeordnet. In der Think-Phasen-Ansicht tippt jede Schülerin und jeder Schüler das eigene Arbeitsergebnis ein und schickt es spätestens am Phasenende ab. Zu Beginn der Pair-Phase trifft sich jede Farbgruppe am Rechner des Mitglieds A, wo anhand der Pair-Phasen-Ansicht (entspricht dem Papier-Placemat) die einzelnen Ergebnisse diskutiert werden und das gemeinsame Gruppenergebnis durch Copy&Paste beschleunigt erstellt und gespeichert wird (s. Abb. 2).

<p>Lisa: Definition für Informatik: Da geht es um Daten und Computer und wie man mit ihnen arbeitet.</p>	<p>Das Ergebnis unserer Gruppe lautet: Definition Informatik:</p>
<p>Florian: Informatik ist die Wissenschaft von der EDV</p>	<p>Informatik ist die Wissenschaft von der systematischen Verarbeitung von digitalen Daten, insbesondere der automatischen Verarbeitung mit Hilfe von Computern</p>
<p>Anna: Meine Definition: Informatik ist eine Wissenschaftsdisziplin, in der sich alles um Computer und Programme dreht.</p>	
<p>Jan: Definieren Sie den Begriff "Informatik" Ich hab folgendes gefunden: Informatik ist die "Wissenschaft von der systematischen Verarbeitung von Informationen, insbesondere der automatischen Verarbeitung mit Hilfe von Digitalrechnern"</p>	
<p style="text-align: right;"><input type="button" value="Gruppenergebnis sichern"/></p>	

Abb. 2: Online-Placemat mit Think-Phase-Ergebnissen links und Gruppenergebnis rechts.

In der abschließenden Share-Phase stellen eine oder mehrere Farbgruppen ihr jeweiliges Ergebnis am Lehrerrechner vor, indem sie es in der Gesamtansicht aufrufen. Abschließend kann die Lehrkraft alle Gruppenergebnisse in Form eines Zip-Archives herunterladen und den SuS zur Verfügung stellen.

2.2 Die Evaluationszielscheibe

Einsatzzweck, Ablauf und Vorteile der Zielscheiben-Methode wurden z.B. in [Be01] oder [Bp12] bereits ausführlich beschrieben.

Die Nachteile der klassischen Variante der Methode auf Papier/an der Tafel sind:

- Es werden Papier und Klebepunkte benötigt und verbraucht.
- Die Punkte müssen entweder zeitaufwändig sequenziell oder in einer Schülertraube parallel aufgeklebt werden.

- Das Feedback geschieht nicht völlig anonym. Später aufklebende SuS können durch früher aufklebende SuS beeinflusst werden.
- Die Zielscheibe kann aus Platzgründen und aufgrund von Manipulationsgefahr häufig nicht bis zur nächsten Stunde im Klassen- bzw. Rechnerraum verbleiben.

Die Webanwendung Online-Zielscheibe wurde entwickelt, um die Methode Zielscheibe im Klassen- oder Rechnerraum unter Umgehung der genannten räumlichen Probleme und Verbesserung der Nachteile der Papierversion zu realisieren.

Die Lehrkraft legt auf ihrer Startseite der Webanwendung zunächst die zu evaluierenden Aspekte fest. Anschließend erscheint die Zielscheibe und darauf im Laufe der Zeit die Punkte der SuS. Neben jedem Abschnitt wird der entsprechende Aspekt zusammen mit dem Mittelwert und der Standardabweichung seines Feedbacks angezeigt. Die zusätzlich angezeigte Anzahl der SuS, die ihr Feedback abgegeben haben, erlaubt der Lehrkraft, den Fortgang der Feedbackabgabe im Blick zu behalten (s. Abb. 3). Um die Anonymität und einen fairen Ablauf zu gewährleisten, werden die IP-Adressen der teilnehmenden SuS unsichtbar im Hintergrund gesammelt und anhand dieser Sammlung werden versuchte Mehrfachteilnahmen abgewiesen. Die Diskussion der Feedbackergebnisse kann durch erneutes Aufrufen der Zielscheibe jederzeit fortgesetzt werden.

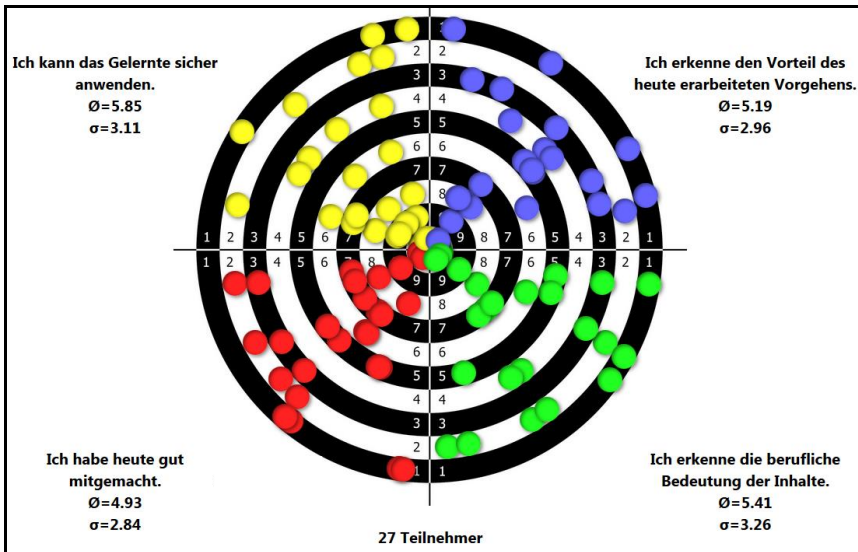


Abb. 3: Online-Zielscheibe mit Evaluationsergebnis von 27 SuS

Die Ansicht der Webanwendung für die SuS besteht aus einer Reihe von Radiobuttons neben jedem zu evaluierenden Aspekt. Die SuS geben ihre Meinung durch die Auswahl der entsprechenden Buttons an und senden anschließend ihr Ergebnis zur Zielscheibe (s. Abb. 4).

	<= ich stimme nicht zu ich stimme voll zu =>			
Ich erkenne den Vorteil des heute erarbeiteten Vorgehens.	1	2	3	4	5	6	7	8	9	10
Ich kann das Gelernte sicher anwenden.	1	2	3	4	5	6	7	8	9	10
Ich habe heute gut mitgemacht.	1	2	3	4	5	6	7	8	9	10
Ich erkenne die berufliche Bedeutung der Inhalte.	1	2	3	4	5	6	7	8	9	10

Abb. 4: SuS-Ansicht mit Evaluationsaspekten und einer zehnstufigen Skala zur Bewertung

2.3 Die Kartenabfrage

Die Kartenabfrage ist eine Methode, die im Unterricht zu unterschiedlichen Zwecken eingesetzt werden kann. Dazu gehören u. a. das Planen weiterer Vorgehensweisen, das Strukturieren von Vorwissen und das Finden von Kompromissen, vgl. z.B. [Ma11] oder [Re07].

Bei der Umsetzung sind einige Hürden und Nachteile zu beachten: Es ist sicherzustellen, dass ausreichend Material zur rechten Zeit am rechten Ort vorrätig ist. Häufig werden die benötigten Materialien in Form von Methodenköffern vorgehalten. Erfahrungen aus der Praxis zeigen jedoch, dass der Einsatz solcher Koffer nicht immer reibungslos funktioniert, insbesondere im Hinblick auf die Aspekte Vollständigkeit und Verfügbarkeit. Das benötigte Papier ist in der Regel nicht wiederverwendbar. Dadurch verursacht jede Kartenabfrage Materialkosten und belastet die Umwelt. Das Ergebnis einer Kartenabfrage ist nur über Umwege (z. B. Abfotografieren oder langwieriges Abschreiben) möglich. Das so gesicherte Ergebnis lässt sich im späteren Verlauf (z. B. in der folgenden Sitzung) nur unter großem Aufwand reproduzieren und weiterbearbeiten. Vor allem in Rechnerräumen ist das Angebot an ausreichend großen Flächen zur Sammlung und Clustering der Karten häufig stark begrenzt⁸.

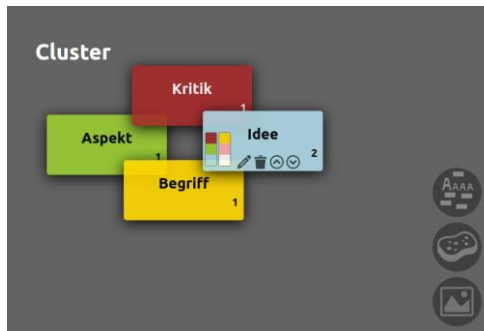


Abb. 5: Clustern und Editieren von Karten

Die Webanwendung Kartenabfrage bietet die Möglichkeit, Karten im Browser eines

⁸ Siehe Kapitel 1.

beliebigen Endgerätes⁹ zu erfassen. Analog zur papierbasierten Kartenabfrage besteht die Möglichkeit, Ideen zunächst zu sammeln und bei Bedarf zu verwerfen. Nach Abschluss der Sammlung können die Karten einzeln oder als Gesamtpaket abgeschickt und damit an die Projektionsfläche gebracht werden (s. Abb. 6). Die Tafelansicht lässt sich an eine geeignete Stelle im Unterrichtsraum projizieren (s. Abb. 5). Am Präsentationsgerät lassen sich die gesammelten Karten beliebig anordnen, editieren und farblich codieren. Die Anzahl der Nennungen der einzelnen Aspekte lässt sich erfassen. Es lassen sich Überschriften ergänzen. Das Ergebnis wird im Hintergrund automatisch gesichert und ist so - über die Angabe eines 4-stelligen Codes - auch zu einem späteren Zeitpunkt editierbar. Die Webanwendung bietet außerdem die Möglichkeit, die Tafelansicht als Bilddatei zu exportieren und so für die Unterlagen zu sichern.

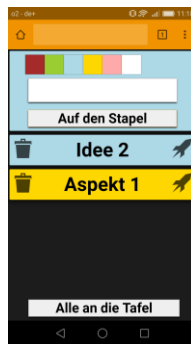


Abb. 6: Erfassen von Karten

2.4 Das Lerntempoduett

Das Lerntempoduett ist eine Methode, bei der SuS Unterrichtsinhalte in mehreren Phasen - zunächst alleine und später in Partnerarbeit – und im eigenen Lerntempo erarbeiten, vgl. z.B. [Ma11]. Die Kombination des Lerntempoduetts mit einem Helfersystem hat sich in der Praxis des Informatikunterrichts bewährt, vgl. z.B. [Re10].

Das Lerntempoduett erfordert – durch seine verschiedenen Phasen und das gewollt ungleichmäßige Vorankommen der SuS – eine gute Methodenkenntnis der SuS und eine wohl durchdachte Begleitung durch die Lehrperson. Die besondere Herausforderung besteht darin, die Vorgehensweise so transparent und eindeutig zu gestalten, dass jedem Schüler und jeder Schülerin zu jedem Zeitpunkt bewusst ist, welche Schritte oder Aufgaben als nächstes zu erledigen sind. Sollen SuS eine helfende Rolle übernehmen, muss den arbeitenden SuS transparent gemacht werden, an wen sie sich wenden können. Aus der Perspektive der Lehrenden ist es im Rahmen des Lerntempoduetts häufig schwierig, das Vorankommen der SuS einzuschätzen und zu beurteilen.

⁹ Siehe Kapitel 2, Tabelle 1.

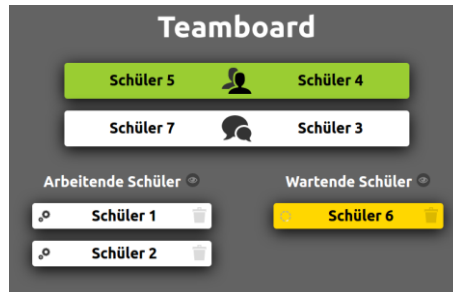


Abb. 7: Tafelansicht des Online-Lerntempoduets

Das Online-Lerntempoduett bietet einen Organisationsrahmen, in dem die SuS schrittweise durch die verschiedenen Phasen des Lerntempoduets geleitet werden und dazu entsprechende Hinweise erhalten. Gleichzeitig wird eine hohe Transparenz hinsichtlich des Vorankommens erreicht. Um das Online-Lerntempoduett zu nutzen, registrieren sich die SuS zunächst in der Webanwendung und werden anschließend durch die verschiedenen Phasen der Methode gelotst. Die Partner für die Partnerarbeitsphase werden vom System automatisch zugewiesen und bekannt gegeben (s. Abb. 8). Das Vorankommen der SuS kann an eine geeignete Stelle im Klassenraum projiziert und damit transparent gemacht werden (siehe Abb. 7). Potenzielle Helfer werden farblich grün hervorgehoben. Um das Verunsicherungspotenzial („naming and shaming“) zu minimieren und die Manipulation der Partnerfindung durch Warten auf einen anderen Partner zu verhindern, lassen sich die SuS in den Phasen „Arbeiten“ und „Warten“ in der Tafelansicht ausblenden.

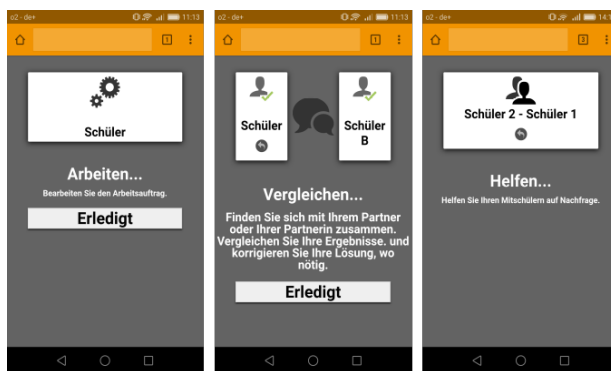


Abb. 8: Verschiedene Phasen des Lerntempoduets. Schüleransichten auf dem Smartphone

3 Ein digitaler Methodenkoffer auf Basis eines Raspberry Pi

Der Einsatz der vorgestellten Webanwendungen setzt voraus, dass die Geräte¹⁰ mit dem Internet verbunden werden. Der Zugriff auf die Anwendungen kann - ohne dass ein Internetzugang von der Schule zur Verfügung gestellt wird - zwar über das mobile Datennetz erfolgen, allerdings wird hierdurch das Datenvolumen der SuS belastet und es entstehen direkte oder indirekte Kosten. Darüber hinaus geht mit dem Einsatz von schülereigenen Smartphones ein nicht zu unterschätzendes Ablenkungspotenzial einher.

Im Rahmen eines Modellprojektes an der BBS Lingen – Agrar und Soziales konnte erfolgreich ein digitaler Methodenkoffer für die Kartenabfrage auf der Basis eines Raspberry Pi, siehe [Ra17a], entwickelt und getestet werden. Der Raspberry Pi dient als WLAN-Accesspoint und Webserver, auf welchem die Anwendung bereitgestellt werden kann. Dazu muss zunächst eine WLAN-Schnittstelle mittels eines entsprechenden USB-Adapters mit Accesspoint-Unterstützung nachgerüstet werden. Nach der Installation muss das System „Raspbian“, siehe [Ra17b], um einen Webserver erweitert werden¹¹. Mittels der Softwaremodule Hostapd und Dnsmasq lässt sich der Raspberry Pi als mobiler WLAN-Hotspot einrichten, siehe [El16], [MF10] und [Me14].

Der Einsatz des digitalen Methodenkoffers auf der Basis eines Raspberry Pi bietet verschiedene Vorteile: Die Betriebskosten und der Anschaffungspreis der eingesetzten Hardware ist gering¹². Das Netzwerk des Methodenkoffers ist unabhängig vom Schulnetzwerk, wodurch keine Gefahr des Missbrauchs der schulischen Infrastruktur besteht. Eine Verbindung zum Internet besteht nicht, wodurch das Ablenkungspotenzial durch Internetdienste verringert wird. Durch die Nutzung des WLAN entstehen keine Kosten für die Nutzung des Mobilfunknetzes. Der Raspberry Pi ist durch seine geringe Größe höchst transportabel im Vergleich zu gängigen Methodenkoffern.

4 Fazit und Ausblick

Die Hoffnung, dass die hier vorgestellten rechnergestützten Realisierungen der vier Methoden die in Kapitel 2 genannten Probleme lösen, hat sich bei der Erprobung durch die Autoren und zusätzlicher Lehrkräfte im Informatikunterricht und im Unterricht in weiteren Fächern an berufsbildenden Schulen in und um Osnabrück erfüllt. Eine ausführliche Darstellung aller beobachteten Aspekte würde den Rahmen sprengen. Zusammenfassend lässt sich sagen, dass die Webanwendungen von den Lerngruppen motiviert genutzt wurden. Sie ließen sich problemlos und gewinnbringend in den Unterrichtsablauf integrieren und die verwendete Technik zeigte sich sowohl in Rechnerräumen als auch in normalen Klassenräumen unter Einsatz von BYOD-Szenarien stabil. Bis heute haben die Webanwendungen etliche Veränderungen in

¹⁰ Siehe Kapitel 2.

¹¹ Im Modellversuch wurde der Apache HTTP-Server genutzt [Ap16].

¹² Unter Einsatz eines Raspberry Pi Zero etwa 25€.

Aussehen und Ablauf erfahren und es ist davon auszugehen, dass diese Entwicklung sich bis zur Tagung fortsetzt.

Literaturverzeichnis

- [Ap16] apache.com: Apache HTTP Server Project. <https://httpd.apache.org/>, Stand: 7.2.2017.
- [Be01] Becker, G.: Qualität entwickeln: evaluieren. Friedrich Verlag, Seelze, 2001.
- [Bp12] Bundeszentrale für politische Bildung (Hrsg.): Auswertungszielscheibe, Evaluationszielscheibe. <http://www.bpb.de/lernen/formate/methoden/62269/methodenkoffer-detailansicht?mid=3>, Stand: 09.02.2017.
- [El16] elinux.org: RPI-Wireless-Hotspot. <http://elinux.org/RPI-Wireless-Hotspot>, Stand: 07.02.2017.
- [GG06] Green, N.; Green, K.: Kooperatives Lernen im Klassenraum und Kollegium, Kallmeyersche Verlagsbuchhandlung GmbH, 2. Auflage, Seelze-Velber, 2006.
- [Ma11] Mattes, W.: Methoden für den Unterricht. Bildungshaus Schulbuchverlage, Paderborn, 2011.
- [Me11] Menzerath, M.: Raspberry Pi als WLAN Access Point nutzen. <https://menzerath.eu/artikel/raspberry-pi-als-wlan-access-point-nutzen/>, Stand: 07.02.2017.
- [MF10] Merkel, H.-P.; Feilner, M.: Mobile WLAN-Accesspoints mit Hostapd. Linux-Magazin 06/2010. <http://www.linux-magazin.de/Ausgaben/2010/06/Macht-Schule>, Stand: 07.02.2017.
- [Ra17a] Raspberry PI Foundation: Raspberry PI Model B. <https://www.raspberrypi.org/products/model-b/>, Stand: 03.02.2017.
- [Ra17b] Raspberry PI Foundation: Downloads. <https://www.raspberrypi.org/downloads/>, Stand: 07.02.2017.
- [Re07] Reich, K.: Unterrichtsmethoden im konstruktiven und systemischen Methodenpool - Brainstorming. www.methodenpool.uni-koeln.de/download/brainstorming.pdf, Stand: 10.02.2017.
- [Re10] Reich, K.: Unterrichtsmethoden im konstruktiven und systemischen Methodenpool – Partnerarbeit. <http://methodenpool.uni-koeln.de/download/partnerarbeit.pdf>, Stand: 10.02.2017.
- [Re14] Reich, K.: Unterrichtsmethoden im konstruktiven und systemischen Methodenpool - Placemat-Methode. [www.methodenpool.uni-koeln.de/download/placemat.pdf](http://methodenpool.uni-koeln.de/download/placemat.pdf), Stand: 04.02.2017.
- [St14] Stommel, J.; Morris, S. M.: Tools for Collaborative Writing, www.jessestommel.com/tools-for-collaborative-writing, Stand: 04.02.2017.

Workshop: Musikprogrammierung mit Sonic Pi

Esther Alzate Romero¹, Leonore Dietrich²

Abstract: Die Musikprogrammiersprache Sonic Pi³, die an der University of Cambridge speziell für Kinder und Jugendliche entwickelt wurde, eröffnet einen kreativen, kontextbasierten Zugang zur Informatik. Ganz nach dem Motto „Technik zum Anfassen“ können durch den Einsatz der günstigen Hardwarelösung Raspberry Pi Berührungsgänge zu Informatiksystemen abgebaut werden. Mithilfe des Design-Based-Research-Ansatzes wird eine Unterrichtseinheit zum Thema Musikprogrammieren unter gendersensiblen Aspekten entwickelt und untersucht. In diesem Workshop werden vor allem die Unterrichtseinheit zum Einstieg in die Programmierung sowie erste Ergebnisse aus der Praxisphase, die daraus resultierenden Erkenntnisse und damit verbundenen Verbesserungs- bzw. Überarbeitungsschritte thematisiert. Hierbei stehen vor allem die Gestaltungsaspekte der Unterrichtseinheit im Vordergrund, ebenso die Frage, welche zentralen Inhalte und Konzepte der Informatik in diesem Rahmen vermittelt werden können. Die Workshopteilnehmer_innen erhalten einen Einblick in die Umgebung und Sprache Sonic Pi, den Unterrichtsgang sowie Materialien und die praktische Umsetzung in der Schule. Zum Abschluss wird ein Ausblick auf die empirische Studie gegeben.

Keywords: Informatikdidaktik, Genderforschung, Unterrichtsforschung, Musikinformatik, Live Coding, Sonic Pi, Design-Based-Research, ProgrammierEinstieg, Informatik im Kontext, Raspberry Pi

1 Einleitung

Der Workshop stellt eine Unterrichtseinheit vor, die im Rahmen eines Dissertationsvorhabens zur Entwicklung und Untersuchung einer gendersensiblen Einführung in die Programmierung entwickelt und erprobt wurde.

Der Zugang zur Programmierung über eine Musikprogrammiersprache soll insbesondere Mädchen ansprechen. Die Durchführung im Musikunterricht wurde gewählt, da Pflichtunterricht Informatik in der Mittelstufe im Evaluationsgebiet nicht existiert. Die Erprobung im Rahmen einer Arbeitsgemeinschaft oder Wahlunterricht hätte dem Ziel, insbesondere Mädchen für Programmierung zu begeistern, widersprochen. Darüber hinaus bestätigen erste Erfahrungen die Erwartung der Autorinnen, dass die Einbettung in einen künstlerischen Kontext mit geringeren Berührungsgängen nicht-technikaffiner Schüler_innen einhergeht.

¹ Heidelberg School of Education, Voßstraße 2, Geb. 4330, 69115 Heidelberg, romero@heiedu.ph-heidelberg.de

² Universität Heidelberg, Didaktik der Informatik, INF 205, 69120 Heidelberg, leonore.dietrich@uni-heidelberg.de

³ <http://sonic-pi.net/> [15.1.2017]

Da es im Wesentlichen um informatische Konzepte geht, die an einem musikalischen Kontext erarbeitet werden, ist die informatische Fachlichkeit der Lehrkraft ein wichtiger Faktor. Ideal ist der Unterricht als Teamteaching-Ansatz mit Informatik als Leit- und Musik als Anwendungsfach umgesetzt. Muss auf eine der Fachlehrkräfte verzichtet werden, dann kann die Einheit auch gut von einer Informatiklehrkraft mit musikalischem Interesse umgesetzt werden.

2 Entwicklung der Unterrichtseinheit

Die Unterrichtseinheit wurde fächerübergreifend von den Autorinnen als Musikinformatikerin und Informatiklehrerin gemeinsam mit einer Musiklehrerin entwickelt. Sie orientiert sich am Mittelstufenunterricht des Faches Musik in Baden-Württemberg und ist ab Klasse 8 geeignet, einen ProgrammierEinstieg zu erarbeiten.

Die Fachkonzepte wurden sowohl nach informatischen als auch nach musikalischen und den jeweiligen didaktischen Konzepten aufbereitet und die Fachsprache in der Unterrichtsvorbereitung auf beide beteiligten Fächer abgestimmt.

Die Einheit wurde erstmals in einer achten Klasse im Rahmen des Musikunterrichts erprobt – unterrichtet wurde sie zum Großteil von einer der Autorinnen als Informatiklehrkraft. Die Musiklehrkraft gestaltete dabei die Verankerung im musikalischen Kontext und übernahm kleinere musikalische und musikdidaktische Anteile des Unterrichts. Bei der ersten Erprobung wurden mehrere kleinere Stücke in Sonic Pi umgesetzt. Die ersten Stunden orientierten sich an bekanntem Liedgut, das neu arrangiert wurde.

Da die Schüler_innen den Teil der eigenen Kompositionen am Ende der Unterrichtseinheit als zu kurz, aber besonders motivierend empfanden, wurde dieses Ergebnis in die Anpassung der Einheit für die zweite Praxisphase übernommen und es wird früher zu eigenen Kompositionen übergegangen. Darüber hinaus wurde ab der zweiten Erprobung als Plattform der Raspberry Pi eingesetzt, da auf diesem Sonic Pi zuverlässiger läuft als auf den in den Schulen vorhandenen Windows-Systemen.

Dieser Workshop stellt den aktuellen, stärker an den eigenen Kompositionen orientierten Unterrichtsverlauf vor.

3 Vorwissen der Lernenden

Da die Einheit für den Mittelstufenunterricht entwickelt wurde, waren bei den bisher erprobten Lerngruppen Vorkenntnisse in den Fächern Musik und Physik vorhanden.

3.1 Physik

Die Vorkenntnisse aus dem Physikunterricht im Bereich Akustik sind hilfreich bei dem Verständnis der Parameter einiger Befehle in Sonic Pi, müssen aber nicht vorausgesetzt werden. Sie können ebenso im Rahmen des Unterrichts erarbeitet und auf die Begriffe Amplitude und Phase beschränkt werden.

3.2 Musik

Für die praktische Umsetzung der eigenen Kompositionen in Sonic Pi sind musiktheoretische Grundkenntnisse hilfreich. Notenlesen muss keine Voraussetzung sein, ist jedoch gerade in der Einstiegsphase, in der ein bekanntes Stück in Sonic Pi nachprogrammiert wird, hilfreich. Auch sind Kenntnisse über den Aufbau musikalischer Stücke hilfreich. In der Praxis zeigte sich, dass vor allem Analogien zu Pop- und Rocksongs und der klassischen Bandstruktur für die Schüler_innen hilfreich waren. So konnte beispielsweise die Synchronisation mehrerer Loops auf einen Loop, der das Metrum vorgibt, gut mit der Rolle des Schlagzeugers in einer Band verglichen werden. Sonic Pi eignet sich vor allem für die Umsetzung von loop-basierter, stark rhythmusorientierter Musik wie Techno, House, Drum 'n' Bass etc. oder für experimentelle Noise-Experimente und flächige Ambientmusik. Da sich die Umsetzung der Komposition ab der zweiten Erprobung mit der Vertonung eines Videos oder Bildes beschäftigt, ist es hilfreich, wenn im Musikunterricht bereits frei zu vorgegebenen Themen/Inhalten improvisiert oder komponiert wurde. Auch die Thematisierung von programmatischer Musik und Filmmusik kann hilfreich sein. Wichtig in diesem Kontext ist die Reflexion des eigenen künstlerischen Schaffens. Das gemeinsame Beschreiben, das Sprechen und die Reflexion über die entstandene Musik, das Finden von geeigneten beschreibenden Adjektiven sind wichtig für den künstlerischen Gestaltungsprozess.

3.3 Informatik

Haben die Lernenden bereits Kontakt mit einer Programmiersprache gehabt, so ist davon auszugehen, dass ihnen der Umgang mit Sonic Pi anfangs leichter fällt, die Erprobungsklassen hatten aber vorher keinen Informatikunterricht. Fachwissen in Informatik wird daher nicht vorausgesetzt und die relevanten Konzepte werden alle als neu zu erarbeiten eingestuft.

4 Ablauf

Die Unterrichtseinheit ist in sechs Doppelstunden angelegt und gliedert sich grob in eine Einführungs- und eine Projektphase.

Stunde	Thema	Bemerkungen
1-2	Einstieg Einheit	Geschichte der Computermusik, Live Coding, Unplugged Modul zu ersten Befehlen mit Parametern
3-4	Einstieg Sonic Pi	Synthesizer, Einzelnoten, Parameter (amp, release..), Schleifen, Samples
5-6	Komposition	Vereinfachte Schreibweise mit Modulo-Operation (Datenstrukturen), Syntax-Highlighting, Debugging, Auskommentieren
7-8	Komposition	Variablen, Effekte, Zufallsfunktionen
9-10	Komposition	Freies Arbeiten
11-12	Präsentation	Freies Arbeiten und Präsentation

Tab. 1: Grobplanung der Unterrichtseinheit Sonic Pi

4.1 Einführungsphase

Nach einer kurzen Einführung in die Computermusikgeschichte über zentrale Figuren der Geschichte der Informatik und der elektronischen Musik folgt eine kurze Demonstration und theoretische Einführung in das Live-Coding, das Programmieren und Improvisieren in Echtzeit von elektronischer Musik und audiovisueller Kunst [Gu13], [BS09], [Co03], [Ro15].

Den ersten Kontakt mit Musikprogrammierung erfahren die Schüler_innen dann unplugged: Zunächst müssen sie herausfinden, welche Parameter beim Musizieren an einem Instrument verändert werden können. Über vorgegebene Schablonen schreiben sie dann entsprechende Befehle an ihre Mitschüler, die von diesen an unterschiedlichen Instrumenten interpretiert werden.

Programmiert euch gegenseitig

Spiele c' leise für $\frac{1}{2}$ Sekunde auf dem Klavier.

Tonhöhe, Lautstärke, Dauer, Instrument

Wähle das Instrument _____
Instrument

Spiele _____ für _____ Sekunde(n).
Tonhöhe Lautstärke Dauer

Spiele _____ für _____ Sekunde(n).
Tonhöhe Lautstärke Dauer

Spiele _____ für _____ Sekunde(n).
Tonhöhe Lautstärke Dauer

Abb. 1: Musikprogrammierung unplugged – Vorlage für den Unterricht

Hierbei kommen bereits mehrere zentrale informatische Konzepte zum Einsatz und werden benannt: Befehl oder Anweisung, Parameter, Konstante, Codierung und Interpretation.

Zentrale Aspekte von Sonic Pi können so erfahrbar gemacht werden: schreibt ein_e Schüler_in bspw. den Pseudocode „spiele auf dem Klavier Ton C1 leise für 20 Sekunden“ bleibt ihm/ihr nichts anderes übrig, als die Taste 20 mal hintereinander zu drücken, da traditionelle Instrumente eine durch die Bauart festgelegte Nachschwingzeit haben. Sonic Pi hat diese Begrenzung (außer es ahmt ein traditionelles Instrument nach) nicht. Daher unterscheidet das Programm zwischen der Nachschwingzeit (*release*), also der Klangdauer und der Dauer, bis der nächste Ton anfängt (*sleep*). Die Begrifflichkeiten werden an dieser Stelle eingeführt und gesichert. Ebenso werden die aus dem Physikunterricht meist bereits bekannten Begriffe der Amplitude und Phase eines Klanges besprochen. Durch den spielerischen Umgang mit der vorgegebenen Notation schreiben die Schüler_innen Pseudocode, der sehr nah an der späteren Umsetzung in Sonic Pi ist.

Sonic Pi Musikprogrammierung *n)))*


Midi-Notation

Der Computer braucht eine eigene Darstellung – er codiert Noten. Dabei ist ein verbreiteter Standard die sogenannte **Midi-Notation**.



Beispiel

Bruder Jakob



Aufgaben

1. Notiere für die erste Zeile von Bruder Jakob die Notennamen über die Midi-Werte.
2. Schreibe den ersten Takt von Bruder Jakob in Sonic Pi.
3. Verändere die Werte für die Eigenschaften und notiere, was sich verändert.
 - a. amp
 - b. release
 - c. sleep.
4. Stelle das Lied bis einschließlich der zweiten Zeile in Sonic Pi fertig.
5. Bearbeite dein Lied – spiele mit anderen Synthesizern, Tempi, zusätzlichen Flächen...



Abb 2: Arbeitsblatt Sonic Pi - Codieren von Noten, vom Pseudocode zum Code

In der zweiten Doppelstunde wird der Pseudocode in Sonic Pi-Code übertragen. Wichtig hierfür ist eine kurze Einführung in die MIDI-Notation. Die Schülerinnen und Schüler übertragen zunächst ein einfaches, ihnen bekanntes Stück in Sonic Pi. Zentrale Konzepte wie Befehl, Parameter, Codierung und Schleife werden hierbei thematisiert und zugleich

hörbar gemacht. Sonic Pi arbeitet für die Erzeugung der Töne mit Synthesizern – die Idee eines Synthesizers kann bereits in der Einführungsstunde vorgestellt und hier wieder aufgegriffen werden. Neben analogen Synthesizern (z.B. Korg Monotron⁴) dürfen die Schülerinnen und Schüler auch digitale Synthesizer (demonstrierbar über eine DAW⁵ mit Plugins) ausprobieren und sich so ein Bild von dieser Klangwelt machen. Die Schüler_innen finden den im unplugged-Teil genutzten Parameter Instrument in der Funktion `use_synth` wieder. Zu Beginn wird der Befehlssatz auf einfache Noten mit den Parametern Tonhöhe, Lautstärke, Länge und Metrum beschränkt, die alle bereits in der vorangegangenen Stunde real erlebbar waren.

Um Melodien leichter umzusetzen werden einfache Schleifen in Form von *loops* eingeführt und als informatisches Konzept herausgearbeitet.

Sonic Pi
Musikprogrammierung
π))

Schleifen - Überblick

Zählschleife

```
2.times do
  play 60, amp: 0.7, release: 1
  sleep 1
  play 62, amp: 0.7, release: 1
  sleep 1
  play 64, amp: 0.7, release: 1
  sleep 1
  play 66, amp: 0.7, release: 1
  sleep 1
end
```

Endlosschleife


```
loop do
  use_bpm 120
  play 60, amp: 0.7, release: 1
  sleep 1
  play 62, amp: 0.7, release: 1
  sleep 1
  play 64, amp: 0.7, release: 1
  sleep 1
  play 66, amp: 0.7, release: 1
  sleep 1
end
```

Benannte Endlosschleife mit Synchronisation

```
live_loop :beat do
  use_synth :chipnoise
  play 60, release: 0.1
  sleep 1
end

live_loop :fastbeat, sync: :beat do
  use_synth :bnoise
  play 60, release: 0.01
  sleep 0.25
end

live_loop :baass, sync: :beat do
  use_synth :prophet
  play 30
  sleep 0.75
  play 35
  sleep 0.75
  play 40
  sleep 0.5
end
```



Esther Alzate Romero
Leonore Dietrich

Pädagogische Hochschule &
Universität Heidelberg 2017

Abb 3: Schleifen und ihre Umsetzung in Sonic Pi

Die für das Live-Coding erforderlichen *live_loops* werden im Zusammenhang mit der *sync*-Funktion erarbeitet, da beides dem synchronen Ablauf dient. Bei Einführung eines solchen Konzeptes werden immer die informatische Bedeutung und die Übertragbarkeit in andere Programmiersprachen hervorgehoben. So werden beispielsweise zunächst Zählschleifen (*x.times*) eingeführt – insbesondere *2.times* wird in Musikstücken häufig verwendet, da es der einfachen Wiederholung dient. Das Konzept wird aus informatischer Sicht besprochen und die musikalische Entsprechung thematisiert. Auch die in der Programmiersprache erweiterte Flexibilität für eigene Kompositionen durch eine wählbare Anzahl von Iterationen wird diskutiert. Mit loops werden anschließend Endlosschleifen als Konzept eingeführt. Erst nach dem Einüben dieser zentralen

⁴ <http://www.korg.com/us/products/dj/monotron/> [10.03.2017]

⁵ Digital Audio Workstation

informatischen Konzepte folgt die Sonderfunktion benannter synchronisierter Schleifen als spezielle Live-Coding Funktion. Diese wird auch als Ausprägung einer spezialisierten Programmiersprache vorgestellt.

4.2 Projektphase

In den folgenden Sitzungen werden zentrale Funktionen innerhalb von Sonic Pi erklärt (Samples, Zufallsfunktion, Effekte) und ein eigenes Stück erarbeitet. Dieses kann alleine oder in Zweiertteams erarbeitet werden. Da die vollkommen freie Komposition eines Stückes für manche Schüler_innen in der ersten Erprobungsphase schwierig war, ist eine angeleitete, eher programmatische Komposition ratsamer. Hierfür eignet sich die Vertonung eines Bildes, einer Geschichte oder eines Kurzfilms. Sonic Pi eignet sich für sphärische Klänge und rhythmische Pattern besonders gut, daher ist eine Arbeit mit Klangcollagen statt dem Erstellen eines „Songs“ empfehlenswert. Ob das erarbeitete Stück live performed und im Sinne des Live-Codings mit dem Programmcode improvisiert wird, oder ob ein fertiges Stück „nur“ abgespielt wird, kann zusammen mit den Schüler_innen entschieden werden.

In dieser Phase können Musiklehrer_innen und Informatiklehrer_innen sehr gut im Team-Teaching interagieren. Der/die Musiklehrer_in kann Fragen zu den künstlerischen Gestaltungsaspekten (wie drücke ich das, was ich sagen, will musikalisch aus? Wie kann ich einzelne musikalische Ideen miteinander verbinden?) beantworten. Der/die Informatiklehrer_in kann bei komplexeren Umsetzungsideen innerhalb des Programms helfen, ebenso kann er/sie beim Debugging unterstützen.

Schüler_innen fragen bald nach Schreibweisen für Akkorde und setzen einfache Datenstrukturen für Melodieschreibweisen ein – Sonic Pi bietet hier beispielsweise einen Ring an, der über einen synchronisierenden Zähler durchlaufen werden kann. Komplexere Indexoperationen können mithilfe mehrerer solcher Ringe experimentell erarbeitet und durch das direkte auditive Feedback sehr einfach verstanden werden. Der Umgang mit gestuften Hilfen und dem in der Umgebung eingebauten Tutorial führen die Schüler_innen schrittweise zu immer selbständigerem Erarbeiten eines größeren Befehlsfundus für die Gestaltung ihrer eigenen Kompositionen.

Die letzte Sitzung dient der Präsentation und einer kurzen Reflexion. Die Schüler_innen präsentieren ihre Stücke und beschreiben, wie sie bei der Komposition vorgegangen sind und was sie mit ihrer Musik ausdrücken möchten. Die Mitschüler_innen geben Feedback, gemeinsam wird über die entstandenen Klänge reflektiert. Hierbei hilft die Übung zur Beschreibung elektronischer Klänge aus der ersten Stunde.

5 Erste Durchführungsphase und Erkenntnisse

Die ersten Ergebnisse des Forschungsvorhabens werden ausführlich im Praxisbeitrag

([AD17a], ebenfalls in diesem Band) beschrieben. Die Schülerinnen und Schüler waren hoch motiviert und begeistert während des Projekts, die Pausen wurden regelmäßig vergessen, stattdessen wurde mit Begeisterung am eigenen Stück gearbeitet. Auch Monate nach Ende des Projektes sprachen Schüler_innen begeistert von der Arbeit mit Sonic Pi im Musikunterricht. Insbesondere die Mädchen der Lerngruppe zeigten langfristiges Interesse. Als sie in der Schule Gelegenheit hatten, erneut Musik zu programmieren, nahmen sie diese begeistert an. Grundlegende Konzepte wie Schleifen und Datenstrukturen wurden von ihnen dabei ohne Unterstützung wieder eingesetzt. Diese zentralen Konzepte waren bei den Schülerinnen also trotz der kurzen Projektdauer und langen Pause noch präsent.

Als problematisch stellte sich die Anwendung unter Windows heraus, da Sonic Pi hier teilweise sehr instabil war. Die Autorinnen empfehlen daher die Anwendung mit einem Raspberry Pi, wie sie ab der zweiten Erprobung auch im Projekt umgesetzt wurde. Positiver Nebeneffekt der Nutzung des Raspberry Pis ist der Abbau von Berührungängsten mit neuer Hardware. Ganz nach dem Motto „Technik zum Anfassen“ wird der Umgang mit einem alternativen Hardwaresystem auf kreative Weise eingeführt.

Als schwierig empfanden die Schüler_innen zudem den Umgang mit den Fehlermeldungen in Sonic Pi. Aufgrund dessen wurde im überarbeiteten Unterrichtskonzept das Debugging stärker in den Fokus gerückt. Künftig erhalten Schüler ein Cheatsheet mit den wichtigsten Fehlermeldungen und deren Erklärung sowie Lösungsvorschlägen dazu. Darüber hinaus wird der Umgang mit Fehlermeldungen im Unterricht stärker thematisiert. Dies soll auch zu einer weiteren Festigung des Verständnisses der informatischen Aspekte führen. Trotz dieser Schwierigkeiten waren die Schüler_innen bis zum Projektende hoch motiviert.

Aus musikalischer Sicht schufen die Schüler_innen begeisternde, komplexe und vielschichtige Klangerlebnisse mit einer abwechslungsreichen Auswahl an Klängen: von Techno bis hin zu sphärischen, naturnahen Kompositionen.

6 Workshopinhalt

Vorkenntnisse in Sonic Pi sind für diesen Workshop nicht erforderlich. Zunächst geben wir eine kurze Einführung in das Feld der Musikinformatik und des Live-Codings. In Anlehnung an den Verlauf der von den Autorinnen konzipierten Unterrichtseinheit werden im Workshop grundlegende Funktionen der Programmiersprache Sonic Pi gemeinsam Schritt für Schritt erarbeitet. Zunächst wird anhand eines bekannten, einfachen Stückes das Schreiben von Melodien erarbeitet. Danach wird eine kleine Klangcollage erstellt, bei der vor allem das Benutzen von Samples (im Programm enthaltene Klangaufnahmen) im Vordergrund steht. Die von den Autorinnen erstellten Arbeitsmaterialien für den Schulunterricht (7.-10.Klasse, alle Schultypen) werden außerdem kurz vor- und den Teilnehmern in der aktuellen Fassung zur Verfügung

gestellt. Anmerkungen und Verbesserungsvorschläge sind sehr willkommen. Nach Abschluss des Forschungsvorhabens werden sämtliche Unterrichtsmaterialien zusammen mit Stundenverlaufsplänen und anderen Handlungsempfehlungen veröffentlicht.

Den Abschluss bildet eine Diskussion der Unterrichtsvorschläge mit Fokus auf den informatischen Wert und die unterrichtliche Umsetzbarkeit.

7 Ausblick: weitere Erprobung und Forschungsvorhaben

In der zweiten und dritten Durchführungsphase wird das überarbeitete Konzept erprobt und erneut verbessert und angepasst. Ebenfalls wird ein abschließender Wissenstest entworfen. Daran schließt sich ein weiterer Forschungsteil an, in dem die gendersensible Unterrichtsgestaltung thematisiert und reflektiert wird. Nach ausführlicher Datensichtung und -auswertung wird die Unterrichtseinheit final überarbeitet.

Literaturverzeichnis

- [AD17a] Alzate Romero, E.; Dietrich, L.: Musikprogrammierung mit Sonic Pi. Entwicklung und Untersuchung einer gendersensiblen Unterrichtseinheit zum Programmieren in der Sekundarstufe I. In (Ira Diethelm, Hrsg.): Informatische Bildung zum Verstehen und Gestalten der digitalen Welt, Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017.
- [BS09] Brown, A. R.; Sorensen, A.: Interacting with Generative Music through Live Coding. *Contemporary Music Review*, 28(1), S.17–29, 2009.
- [Co03] Collins, N.; McLean, A.; Rohrhuber, J.; Ward, A.: Live coding in laptop performance. In: *Organised Sound*, 8(3)/03, S.321–329, 2003.
- [Gu13] Guzdial, M.: Live-Coding in Education. In (Blackwell, A.; McLean, A.; Noble, J.; Rohrhuber, J., Hrsg.): *Collaboration and learning through live coding (Dagstuhl Seminar 13382)*. Dagstuhl Reports, 3/13, S. 135-136, 2013.
- [Ro15] Rohrhuber, J.: Lehre als Forschung: Grundlagen der Musikinformatik im künstlerischen Kontext. *Zeitschrift für Hochschulentwicklung*, 10(1)/15, S.79-91, 2015.

Herausforderungen durch neue Programmierkonzepte in blockbasierten Programmiersprachen

Sven Jatzlau¹, Ralf Romeike²

Abstract: Blockbasierte Programmiersprachen stehen für einen Zugang zur Programmierung, der von Programmieranfängern erfolgreich genutzt wird und zunehmend auch als Möglichkeit gesehen wird, nicht-professionellen Programmierern das Gestalten von Informatiksystemen zu ermöglichen. Als Gründe hierfür werden u. a. die intuitive Bedienung, schnelle Erfolgserlebnisse und ein breites und kontextualisiertes Anwendungsspektrum genannt, die zu einer großen Beliebtheit unter jungen Programmierern führen und sich als Kernmerkmale in den populären Programmierumgebungen Scratch und BYOB/Snap! wiederfinden. Diese grafische, blockbasierte Programmierung unterscheidet sich jedoch von klassischer textbasierter Programmierung nicht nur auf der Bedienebene, sondern bringt gegenüber gängigen im Unterricht genutzten Programmiersprachen auch verschiedene konzeptuelle Unterschiede mit sich. Solche ergeben sich zum einen aus der visuellen Darstellung von Objekten auf der Bühne, zum anderen aus der historischen Genese sowie didaktischen und pragmatischen Entscheidungen. Für Lehrerinnen und Lehrer ist ein konzeptuelles Verständnis wichtig, um Möglichkeiten und Probleme beim Programmierenlernen mit Scratch-ähnlichen Programmiersprachen zu erkennen und didaktisch darauf einzugehen. Im vorliegenden Beitrag werden verschiedene zentrale Konzepte, wie z. B. Nesting von Grafikobjekten, herausgearbeitet und verdeutlicht. Es zeigt sich, dass der Ansatz des „Programmierens für alle“ mit passenden Werkzeugen in greifbare Nähe rückt, eine didaktische Aufarbeitung zum Erreichen eines informatischen Grundverständnisses aber unerlässlich ist.

Keywords: visuelle Programmierung, Scratch, Snap!, BYOB, GP, blockbasierte Programmiersprachen

1 Motivation und Ziele

Blockbasierte Programmierung ist spätestens seit *Scratch* ein relevantes Thema für Programmieranfänger, ob in Schule oder Freizeit. Programmiersprachen, die sich auf das Konzept der blockbasierten Programmierung stützen, unterscheiden sich von textbasierten Sprachen sowohl auf konzeptioneller Ebene als auch in ihrer Interaktion mit dem Benutzer. Aufgrund ihrer Einfachheit und niedrigen Einstiegshürde (u.a. durch wegfallende Syntaxfehler) sind besonders die blockbasierten Sprachen, zu denen etwa *Scratch* und *Snap!* gehören, unter Anfängern und Einsteigern beliebt (vgl. [LKG17]).

Aber auch abseits der Syntaxreduzierung wurden weitere Entscheidungen getroffen, die das Verständnis der Schüler fördern: Im Folgenden soll herausgearbeitet werden, welche

¹ Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, Germany sven.jatzlau@fau.de

² Friedrich-Alexander-Universität Erlangen-Nürnberg, Didaktik der Informatik, Martensstr. 3, 91058 Erlangen, Germany ralf.romeike@fau.de

Unterschiede blockbasierte Programmiersprachen mit sich bringen und welche neuen Konzepte in diesem Zusammenhang wichtig sind. Zu diesem Zweck werden zunächst die historischen Ursprünge blockbasierter Programmierung analysiert und dann die wichtigsten neuen Konzepte anhand von Sprachen wie *Scratch*, *Snap!* und einem neuen Vertreter der Familie, *GP*, verdeutlicht. Die Konzepte nichtatomarer Interpreter, Klassen und Objekte, Nested Sprites, First-class-Strukturen, Entwickler-Modus, Block-to-Text-Slider und Class Browser wurden ausgewählt, da sie entweder in ihrer Darstellung neuartig oder aber für Neulinge in der blockbasierten Programmierung noch gänzlich unbekannt sind, weshalb es wichtig ist, dass sie didaktisch behandelt und aufgearbeitet werden.

2 Forschungsstand

Zum Thema Didaktik der blockbasierten Programmierung wurde bis heute nur relativ wenig Forschungsarbeit geleistet. Mitchel Resnick, verantwortlich für die Entwicklung der Sprache *Scratch*, kritisierte bereits 2002, dass Computer zwar einerseits immer zugänglicher und verbreiteter werden und dadurch eine Revolution der Lerngewohnheiten möglich ist, aber oftmals digitale Medien nur dazu verwendet werden, veraltete Herangehensweisen an den Lernprozess zu unterstützen [Re02]. Resnick et al. [RMMH09] machen darauf aufmerksam, dass digitale Kompetenz nicht nur Interaktion und Kommunikation beinhalten darf, sondern auch das Erschaffen und die Weiterentwicklung von Inhalten. Durch Modrow et al. [Mo11, MMS11] wurden die Einstellungen von Schülern zu *Scratch* untersucht. Dabei war klar erkennbar, dass Sprachen wie *Scratch* an Schulen zwar gerne von ihnen verwendet werden, jedoch als reine Einsteigerprogramme verstanden werden. Für komplexere Aufgabenstellungen oder im Berufsleben wird *Scratch* etwa als nicht angemessen empfunden. Gleichzeitig jedoch sind beispielsweise alle untersuchten Abituraufgaben (Niedersachsen, 2011) problemlos in *Scratch* oder *Snap!* lösbar, sodass eine Abwendung von textbasierter Programmierung hin zu blockbasierter durchaus denkbar wäre. Die Forschungsergebnisse von Strecker [St15] unterstützen diese These: Sie verglich die Leistungen von Schülern, die mit *Snap!* auf das Abitur vorbereitet wurden, mit denen, die reine Java-Kurse besuchten. Im Durchschnitt erbrachten Erstere bessere Leistungen, v.a. im Hinblick auf Teillösungen. Außerdem waren sie den Herausforderungen besser gewachsen, da ihnen weniger Konzepte oder Kompetenzen fehlten. Besonders im Bereich der Algorithmik konnten dadurch bessere Ergebnisse erzielt werden. Price et al. [PB15] erforschten das Verhalten von Schülern bei der Verwendung blockbasierter Programmiersprachen. Laut ihren Forschungen waren diese Schüler in der Lage, Programmieraufgaben deutlich schneller zu lösen als solche, die textbasierte Programmiersprachen verwendeten. Dieses Ergebnis deutet darauf hin, dass blockbasierte Programmiersprachen und Benutzeroberflächen leistungsfördernd wirken. Dies könnte mit der Motivation der Lerner zusammenhängen: Ruf et al. [RMH14] verglichen intrinsische Motivation und Programmierfähigkeiten von zwei Gruppen von Lernern. Eine Gruppe verwendete *Scratch* zum Erlernen der Programmierung, die andere die textbasierte Programmiersprache *Karol*. Hierbei zeigte sich, dass sowohl Motivation als auch Programmierkompetenz in der *Scratch* Klasse stärker vertreten waren. Die Forschungsergebnisse von

Weintrop et al. [WW15] zeigen ähnliche Ergebnisse: Schülern wurden Aufgaben vorgelegt, die sie bearbeiten sollten. Der zu bearbeitende Code wurde jeweils in Blöcken bzw. in Textform vorgelegt, doch die blockbasierten Aufgaben wurden deutlich häufiger korrekt beantwortet. Dwyer et al. [DHH15] erforschten, wie Informationen in blockbasierten User-interfaces aufgenommen und verarbeitet werden und zeigten, dass die visuelle Natur solcher Programmierumgebungen oftmals hilfreich für Lerner ist. Trotzdem besteht die Gefahr, dass Nutzer Funktionen der Umgebung übersehen oder visuelle Hinweise falsch interpretieren (etwa wenn die Position oder das Aussehen der Objekte auf der Bühne fälschlicherweise als Hinweise für ihr Verhalten gedeutet werden). Aber auch auf motivationaler Ebene zeigen sich nicht ausschließlich Vorteile: Lewis [Le10] untersuchte die Wahrnehmung von Lernern bei der Verwendung von *Scratch* und *Logo*. Zwischen den Gruppen zeigten sich keine bedeutenden Unterschiede hinsichtlich ihrer Motivation, zukünftig zu programmieren.

Es wird deutlich, dass blockbasierte Programmiersprachen besonders für Programmieranfänger eine große Chance darstellen, schnell und sicher Kompetenzen im Bereich der Algorithmik und Programmierung zu erlangen. Dies wirkt sich positiv auf deren intrinsische Motivation und Selbsteinschätzung aus.

3 Meilensteine in der Entwicklung der blockbasierten Programmiersprachen

Ein großer Teil der konzeptionellen Neuerungen, die sich in blockbasierten Programmiersprachen finden, lassen sich auf die historische Entwicklung zurückführen. Daher ist es bedeutsam, die Meilensteine in der Entwicklung dieser Programmiersprachenfamilie zu kennen. Hierzu sind drei essentielle Programmierumgebungen als Meilensteine zu nennen, die neben *Scratch* und *Snap!* die Genese visueller Sprachen beeinflussen: *Morphic*, *Smalltalk* und *GP*. Diese sollen im Folgenden kurz erläutert und ihre Bedeutung für die blockbasierte Programmierung herausgestellt werden. Diese Umgebungen wurden gewählt, da sich ihre Konzepte schließlich in den aktuellen Programmierumgebungen wiederfinden, sodass sie ihre Entwicklung maßgeblich beeinflussten bzw. Ansätze der zukünftigen Weiterentwicklung aufzeigen (*GP*).

Morphic ist ein Framework, das es dem Benutzer ermöglicht, auf einfache Art und Weise ein grafisches Userinterface zu erstellen und mit diesem zu interagieren. Das Framework basiert dabei sowohl auf einer direkten Manipulation der jeweiligen Objekte (anhand von Kontextmenüs) als auch einer Manipulation innerhalb von Programmen. Ursprünglich wurde es als Teil der Sprache Self bei Sun entworfen, die erstmals in den 80ern und 90ern als experimentell verwendetes Testsystem für Sprachendesign eingesetzt wurde. Grundlegend für diese Umgebung ist, dass alle sichtbaren Objekte sog. „Morphs“ sind und somit etwa auf Ereignisse reagieren (wie etwa Mausklicks), eine Überlappung mit anderen Objekten erkennen oder aus Teilobjekten bestehen können [SWM13]. Dadurch besteht die Möglichkeit, Objekte zu verschachteln, sodass diese in einer Teil-Ganzes-Beziehung stehen, wodurch jeder „Morph“ eine bestimmte Stelle in einer Hierarchie einnimmt (an deren Spitze

die „World“ bzw. die Bühne steht). Einige der Konzepte finden sich etwa in *Scratch* wieder: So dürfte das Prinzip der Verschachtelung als Vorlage für „Nested Sprites“ in *Snap!* gedient haben.

Smalltalk ist das Resultat langwieriger Forschung, deren Ziel es war, dem Benutzer die Interaktion mit einem Computer auf einer funktionalen Ebene möglich zu machen [GR83, 8]. Es handelt sich hier um eine objektorientierte Programmiersprache, die gezielt zum Einstieg in die Programmierung konzipiert wurde. Wichtig war laut Entwickler Alan Kay die Bereitstellung einer Umgebung, in der Erkundung belohnt wird (nach Montessori), die enaktives, ikonisches und symbolisches Lernen ermöglicht und fördert (nach Piaget und Bruner), in der die Magie im Bekannten steckt (Negroponte) und die als verstärkender Spiegel für die Intelligenz des Benutzers dient (Coleridge, [Ka96, 33]). Die Sprache basiert auf dem Kernkonzept „Alles ist ein Objekt“. Ausgehend von diesem Konzept gelten bestimmte Regeln für Objekte, wie etwa, dass sie untereinander mithilfe von Nachrichten (die Objekte sind) kommunizieren oder, dass Objekte über ein eigenes Gedächtnis verfügen (das wiederum ein Objekt ist). *Smalltalk* und seine spätere Weiterentwicklung, bzw. Modifikation *Squeak* sind in ihrer Entwicklungslaufbahn zur Basis von diversen visuellen Programmierumgebungen geworden, wie z. B. *Scratch* oder *Snap!*. Die objektorientierte Natur dieser Sprache, in der mit grafischen Objekten anhand von Nachrichten kommuniziert wird (etwa eine Nachricht an eine Box, sich zu drehen oder zu vergrößern) wurde als sehr motivierend aufgefasst, sodass bereits Kinder in der Lage waren, Malprogramme oder Notenblatt-Lesesysteme zu entwerfen. Diese Tatsache könnte ein Grund dafür sein, dass sich das Prinzip des Nachrichtenversands auch in *Scratch* oder *Snap!* wiederfindet.

GP ist der neueste Vertreter der Sprachenfamilie von *Scratch* und befindet sich momentan noch in der Entwicklungsphase. Es wird derzeit u.A. von Jens Mönig und John Maloney entwickelt und basiert auf der Vision, dass dem Benutzer weniger Grenzen vorgegeben werden sollen als etwa in *Scratch*. Das bedeutet, dass komplexe Programmierkonzepte, die in den verwandten Sprachen aus didaktischen Gründen fehlen, umsetzbar sein sollen. Außerdem war es das Ziel, eine erweiterbare Plattform zu erschaffen - der interne Code der gesamten Programmierumgebung ist einsehbar und manipulierbar, sodass Modifikationen inhärent unterstützt werden. Die Programmierumgebung ist *Scratch* und *Snap!* sehr ähnlich, weist jedoch einige Unterschiede auf: Sie ist beinahe vollständig in sich selbst implementiert, d.h. wenn Codeblöcke untersucht werden, um deren interne Funktionsweise sichtbar zu machen, enthüllt sich wiederum Code aus grafischen Blöcken. Dieses Konzept findet sich durchgehend bis zur untersten Ebene, sog. „primitives“, also primitiven Funktionen, die nicht weiter einsehbar sind, da sie in C implementiert sind. Neben diesem Konzept finden sich in *GP* jedoch noch weitere konzeptionelle Neuerungen: Wie in den anderen Vertretern der Sprachenfamilie verfügt es über einen nichtatomaren Interpreter. Außerdem ist es in der Lage, „Nested Sprites“ zu erschaffen, mit Dateisystemen zu interagieren und grafische Codeblöcke als Text darzustellen. Alle diese Phänomene werden im folgenden Kapitel näher erläutert.

4 Neue Konzepte blockbasierter Programmierung

Im Folgenden sollen nun fünf der wichtigsten konzeptionellen Unterschiede und Neuerungen aufgezeigt werden. Die Konzepte sind bedeutsam, da sie klare Neuerungen darstellen, die durch visuelle, blockbasierte Programmiersprachen eingeführt wurden: Sie existieren in dieser Form nicht in den bekannten textbasierten Programmierumgebungen, die momentan noch etwa in der Schule Verwendung finden. Daher ist es wichtig, dass Lehrer sich dieser Neuerungen bewusst sind, sodass sie didaktisch behandelt werden können. Diese Konzepte finden sich in den aktuellen Vertretern blockbasierter Programmierumgebungen: *Scratch*, *Snap!* und *GP*.

Nichtatomare Interpreter und Debugging Die wichtigsten Neuerungen haben eins gemeinsam: Sie sind auf didaktische Gründe zurückzuführen, d.h. die Motivation war es, die Programmiersprache intuitiver zu gestalten. Hierzu gehört z. B. die Eigenschaft, dass Codeblöcke verzögert ausgeführt werden. Das bedeutet, dass bestimmte Skripte langsamer ausgeführt werden als die Umgebung (und der Computer) es eigentlich zulassen würde(n). Ohne diese Verzögerung könnten z. B. Sprites sofort nach Programmstart die Grenzen der Leinwand verlassen, ohne dass für den Benutzer ersichtlich wäre, warum. Um Anfänger vor diesem Phänomen zu bewahren, wurde die Verzögerung für bestimmte Block-Arten eingeführt: Alle Schleifen-, Warte- und Bewegungsblöcke haben diese Verzögerung. Diese Verzögerung kann etwa in *Snap!* mithilfe zweier verschiedener Maßnahmen umgangen werden: Entweder durch die Verwendung eines speziellen „Warp“-Blocks oder durch Aktivierung des „Turbo“-Modus.

Dieser Unterschied in der Ausführung des Codes ist ein wichtiges Konzept, das für fortgeschrittene Lerner u.U. zu großer Verwirrung führen kann. Daher ist es wichtig, bei der Vermittlung der Programmierung auf die neuen Anforderungen einzugehen. Besonders für Debugging ist das Konzept der verzögerten Ausführung nützlich: *Snap!* bietet beispielsweise die Möglichkeit, die Ausführungsgeschwindigkeit von Blöcken manuell zu regeln. Die Option „visible stepping“ erweitert die Benutzeroberfläche um einen Schieberegler, der die Ausführungsgeschwindigkeit des Codes stufenlos reduziert, sodass der Programmablauf einfach verfolgt und nachvollzogen werden kann. Der Verstehensprozess wird zusätzlich durch Hervorhebung des momentan ausgeführten Codeblocks unterstützt. Hier wird deutlich, wie die Visualisierung des Codes (nicht nur auf der Leinwand, sondern auch in der direkten Darstellung) verständnisfördernd aufbereitet werden kann.

Klassen-/Objekt-Darstellung Auch die Darstellung von Klassen und Objekten (als Instanzen von Klassen) unterliegt einigen Änderungen im Kontext der blockbasierten Programmierung: Hier wird der „Prototyping“-Ansatz verwendet, nach dem Objekte („Sprites“) geklont werden und dadurch neue Objekte erzeugt werden, die ihrerseits exakte Kopien sind und alle Codeblöcke und Attribute des ursprünglichen Objektes übernehmen. Dadurch „beschreibt“ der Benutzer ein Beispiel einer Instanz, keinen abstrakten Bauplan (Klasse). Dieser Prototyp kann dann weitere Instanzen von sich selbst erzeugen, die seinen Code teilen. Diese Form der Darstellung wurde von der Arbeit von Henry Lieberman inspiriert,

der verdeutlichte, dass durch das Prinzip von Prototypen „Standards“ und Abweichungen von solchen intuitiv verständlich sind [Li86]. Der verständnisfördernde Charakter des Prototypen-Konzepts wurde als didaktisch sinnvoll erfasst, sodass es in vielen blockbasierten Programmiersprachen, wie etwa *Scratch* und *Snap!*, Verwendung findet. Modrow zeigt auf, wie hier das objektorientierte Konzept der Vererbung über das Delegations-Modell realisiert werden kann [Mo13]. Prinzipiell sind zwei Arten des Klonens zu unterscheiden: Das Klonen zur Laufzeit (durch Ausführung der entsprechenden Codeblöcke) und das Klonen zur Programmierzeit (durch „duplicate“-Befehl im Kontextmenü eines Objekts). Ersteres erschafft eine exakte Kopie des Originals, die bei Programmende entfernt wird, während Letzteres einen permanenten Klon erschafft, dessen Aussehen und Verhalten danach verändert werden können ohne das ursprüngliche Objekt zu beeinflussen (Abb. 1).

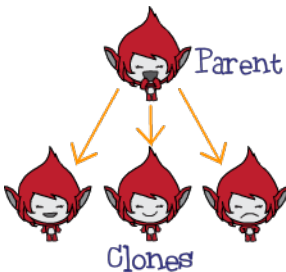


Abb. 1: Visuelle Darstellung des Klon-Konzepts in *Scratch* [Sc17a]

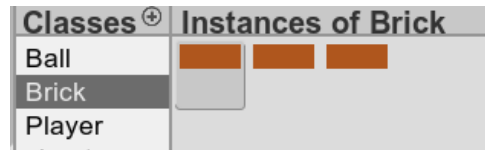


Abb. 2: Klasse und ihre Instanzen in *GP*

Auch *GP* folgt der Idee des Prototyping, greift aber für die Darstellung auf das von Smalltalk bekannte und aus der textbasierten Programmierung verbreitete Klasse-Objekt-Konzept zurück, wie es beispielsweise in Java oder C++ implementiert wird. Die Abb. 2 ist ein Ausschnitt aus der Programmieroberfläche. Es ist eine Auflistung aller Klassen zu sehen, die im laufenden Projekt Verwendung finden und rechts davon eine Übersicht über die Instanzen dieser Klasse (und ihrem momentanen Aussehen). Für beide Bereiche gibt es +-Knöpfe, deren Betätigung eine neue Klasse bzw. eine Instanz von ihr erzeugt. Objekte derselben Klassen teilen sich alle den Code, der für diese Klasse definiert wurde - sie reagieren auf dieselben Ereignisse und führen dieselben Codeblöcke aus. Jedes Objekt ist dadurch die grafische Instanz einer Klasse. Wird eine neue Klasse erzeugt, so wird gleichzeitig die erste Instanz dieser Klasse auf der Leinwand erschaffen, die somit als Prototyp fungiert. Existiert nur eine Instanz pro Klasse, so ist das Verhalten der Programmierumgebung funktional ähnlich zu *Scratch* – mit dem Unterschied, dass eine explizite Auswahl des Klassennamens aus der Liste (vgl. Abb. 2) notwendig ist, um die konkrete Instanz anwählbar zu machen. Werden von einer Klasse mehrere Instanzen erzeugt, machen sich die Unterschiede zu *Scratch* und *Snap!* bemerkbar: In *GP* bezieht sich der Code immer auf eine ganze Klasse. Wird dieser verändert, so ändert er sich für alle Instanzen dieser Klasse (analog zum Laufzeit-Klonen in *Scratch/Snap!*). *Scratch* erlaubt hingegen zusätzlich, nach dem Klonen den Code eines Objektes zu ändern, ohne dabei den anderer Klone oder des „Originals“ zu beeinflussen (durch Klonen zum Programmierzeitpunkt).

Es wird deutlich, dass die Frage, in welcher Form das Klassen/Objekt-Prinzip vermittelt werden soll, zentral ist. Die Unterschiede in der Darstellung von Vererbung und Klassen/Objekten müssen für eine didaktisch wertvolle Vermittlung beachtet und reflektiert werden.

„Nested Sprites“ Das Konzept der „Nested Sprites“ ist ein gutes Beispiel dafür, dass visuelle Programmiersprachen neue Problemlöseansätze eröffnen (die im Unterricht thematisiert werden müssen). Es handelt sich hierbei um eine örtliche Verbindung von Objekten bzw. von „Sprites“ auf der Leinwand: Sie stehen in einer Teil-Ganzes-Beziehung zueinander und nehmen dadurch einen Platz in einer Objekthierarchie ein. Ein Objekt kann aus mehreren Teilen bestehen und selbst Teil eines Objektes sein. Die Rotationen und Bewegungen der beteiligten Objekte sind demnach voneinander abhängig (bewegt sich der Besitzer, so bewegen sich alle seine Teilobjekte mit ihm).

Das Konzept ist charakteristisch für visuelle Sprachen, da es nur durch die visuelle Repräsentation von Objekten möglich gemacht wird (gäbe es keine Leinwand, um Objekte zu visualisieren, gäbe es auch das Konzept der verschachtelten Objekte nicht). Es kann beispielsweise verwendet werden, um Simulationen zu implementieren, wie etwa die mechanische Funktionsweise eines Verbrennungsmotors: Ein Teil bewegt sich, wodurch sich wiederum andere Teile bewegen, sodass schließlich eine Verbrennung stattfindet (Abb. 3).



Abb. 3: Vereinfachte Simulation eines Verbrennungsmotors aus GP

In *Snap!* kann eine solche Beziehung zwischen Objekten durch Drag-and-drop erzeugt werden, sodass Objekte an andere Objekte angefügt werden. In *GP* muss hierzu ein Objekt mit einem Anderen überlappen, woraufhin per Kontextmenü eine Zuweisung „attach“ verfügbar wird.

„First class Strukturen“ Smalltalk, Morhic, *Snap!* und *GP* setzen ein aus der funktionalen Programmierung bekanntes und mächtiges Konzept um, das durch die visuelle Programmierung intuitiv verständlich und damit gut handhabbar umgesetzt wird: Alle vorhandenen Strukturen und Objekte sind „first class“. Strukturen und Objekte werden als „First class objects“ bezeichnet, wenn sie folgende Eigenschaften erfüllen (vgl. [HM17]):

- sie können Werte von Variablen sein,
- sie können Parameter und Rückgabewerte von Funktionen sein,
- sie sind anonym (werden bei Erstellung nicht benannt),
- sie können Teil eines Aggregats sein.

Geprägt wurde der Begriff von Christopher Strachey, der um 1960 forderte, dass jeder

Datentyp einer Sprache „first class data“ sein sollte [Sc17b], sodass jeder Datentyp völlig uneingeschränkt benutzbar ist. Dies ist eines der fundamentalen Prinzipien von *Snap!*. Durch die daraus resultierenden Anwendungsmöglichkeiten (wie etwa Listen von Listen) bieten sich neue Herangehensweisen und Lösungsmöglichkeiten für bekannte Probleme an. Beispielsweise können mapping-Funktionen über Listen durchgeführt werden oder Blöcke als Eingabeparameter für andere Blöcke fungieren.

Entwicklermodus, Block-to-Text-Slider, Class Browser Diese Konzepte stehen bislang nur in der Umgebung *GP* zur Verfügung. Der Entwicklermodus stellt eine optional zuschaltbare Erweiterung der Funktionalität der Programmierumgebung dar. Bei aktivem Entwicklermodus werden einige Paletten erweitert, wie z. B. Variablen um den Typ *script* oder die „New Class“-Funktion um die „helper“-Variante. Auch der „Class Browser“ wird zugänglich: Hier findet sich der Systemcode von *GP* (näheres hierzu weiter unten).

Die wohl interessanteste Erweiterung offenbart sich als Schieberegler in der Menüleiste: ein stufenloser Regler der Codedarstellung von Blöcken zu Text. Dieser soll die Brücke zu textbasierten Programmiersprachen repräsentieren und die Äquivalenz der beiden Repräsentationen verdeutlichen – je nach subjektiver Präferenz kann der Benutzer seine Bedienung entweder blockbasiert oder textbasiert einstellen (vgl. Abb. 4).

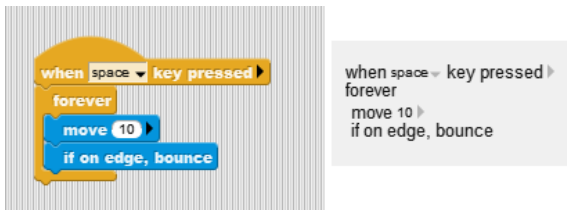


Abb. 4: Beispielcode in Block- und in Textmodus

Fließende Übergänge ergeben sich aus der alternativen Bedienung mit der Tastatur. Wie *Snap!* bietet *GP* die Möglichkeit, Codeblöcke durch Tastatureingaben (mit Autovervollständigung) zu erzeugen: Die Umgebung reagiert auf neu eingegebene Zeichen damit, dass alle verfügbaren Kategorien nach passenden Blöcken durchsucht und in einer Liste dargestellt werden. Der passendste Block wird stets ganz oben in der Liste angezeigt, und eine Bestätigung per Enter-Knopf fügt diesen an die momentan ausgewählte Stelle im Skript an. Die Kategorie, der der angefügte Block angehört, wird zudem direkt angewählt, sodass ähnliche Blöcke sofort sichtbar sind – eine Funktionalität, die *Scratch* nicht bietet. Dadurch ist eine vollständige Bedienung der Oberfläche durch die Tastatur möglich.

Zweifellos ist dieser Eingabemodus, v.a. gepaart mit dem Block-zu-Text-Slider auch als Brücke zu textbasierten Programmiersprachen intendiert. Wenn beide Modi aktiviert sind, gibt die Benutzung das Gefühl, eine textbasierte Programmierumgebung mit starker Syntaxunterstützung zu verwenden (wie beispielsweise *Strides* in *Greenfoot*). Hier zeigt sich erneut, dass *GP* als „Lösung“ dafür erschaffen wurde, dass das visuelle Programmieren

nicht „echt“ erscheint, zu unprofessionell präsentiert wird und nicht wirklich übertragbar auf andere Programmiersprachen scheint.

Das letzte Konzept, der Class Browser, verkörpert die Vision von *GP*, dass selbst komplexeste Projekte realisierbar sind – „high ceiling“ nach Seymour Papert. Im Class Browser sind alle Skripte gelistet, die in der Palette zu finden sind und darüber hinaus alle Skripte, die das interne Verhalten der Umgebung beschreiben. Der Class Browser macht es möglich nachzuvollziehen, wie die Funktionsweisen der Programmierumgebung bis zur untersten Ebene hin implementiert sind (Abb. 5); so lässt sich die Implementierungshierarchie von Skripten bis zur primitiven Ebene hin verfolgen. Nutzbar ist er dadurch beispielsweise, um vordefinierte Skripte zu ändern, ergänzen oder zu löschen und somit eine eigene *GP*-Version völlig den eigenen Ansprüchen entsprechend zu erstellen.

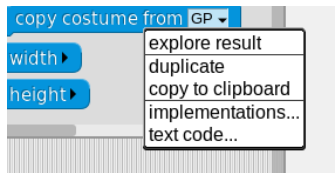


Abb. 5: Verwendung des Class Browsers: „implementations...“

Bei diesen Konzepten ist eine besondere Bedeutung für Schüler sichtbar: Anfangs werden komplexe Funktionen und Blöcke verborgen, sodass eine niedrige Einstiegshürde gegeben ist. Bei aktiviertem Entwickler-Modus wird dann die selbstständige Erkundung der Programmierumgebung und ihres Codes durch den Class Browser möglich, sodass interessierten Schülern kaum Grenzen gesetzt sind. Gleichzeitig ermöglicht der Block-zu-Text-Schalter eine subjektive Regelung der Programmieroberfläche, sodass Lerner sie je nach Präferenz, Kenntnisstand oder Programmierfähigkeit individuell anpassen können.

5 Fazit

Die wachsende Präsenz blockbasierter Programmiersprachen kann für Schüler große Vorteile mit sich bringen. Dennoch birgt diese neuartige Form der Programmierung auch Herausforderungen durch neue Konzepte, die verstanden und erforscht werden müssen. Prinzipien wie der nichtatomare Interpreter, der Skripte verlangsamt ausführt, sodass diese intuitiver ablaufen oder Nested Sprites, die beinahe ausschließlich auf der visuellen Repräsentation von Objekten basieren, sind neu und aus didaktischer Sicht noch praktisch unerforscht. Trotz der intuitiven Natur visueller und insbesondere blockbasierter Programmiersprachen ist hier also noch viel Forschungsarbeit zu leisten - bisher wurden erst Ansätze für eine Didaktik visueller Programmiersprachen geliefert.

Literaturverzeichnis

- [DHH15] Dwyer, H.; Hill, C.; Hansen, A. et al.: Fourth Grade Students Reading Block-Based Programs: Predictions, Visual Cues, and Affordances. In: Proceedings of the eleventh annual International Conference on International Computing Education Research. ACM, S. 111–119, 2015.
- [GR83] Goldberg, A.; Robson, D.: Smalltalk-80: the language and its implementation. Addison-Wesley Longman Publishing Co., Inc., 1983.
- [HM17] Snap! Reference manual online, snap.berkeley.edu/SnapManual.pdf, Stand: 09.06.2017.
- [Ka96] Kay, A.: The early history of Smalltalk. In: History of programming languages—II. ACM, S. 511–598, 1996.
- [Le10] Lewis, C.: How programming environment shapes perception, learning and goals: logo vs. scratch. In: Proceedings of the 41st ACM technical symposium on Computer science education. ACM, S. 346–350, 2010.
- [Li86] Lieberman, H.: Using prototypical objects to implement shared behavior in object-oriented systems. In: ACM Sigplan Notices. Jgg. 21. ACM, S. 214–223, 1986.
- [LKG17] Scratch: Statistics, scratch.mit.edu/statistics/, Stand: 09.06.2017.
- [Mo11] Modrow, E.: Visuelle Programmierung – oder: Was lernt man aus Syntaxfehlern? In (Thomas, M., Hrsg.): Informatik in Bildung und Beruf. Jgg. 14 in Lecture Notes in Informatics (LNI), Köllen, Bonn, S. 27–36, 2011.
- [Mo13] Informatik mit BYOB / Snap!, www.uni-goettingen.de/de/informatik-mit-byob/423680.html, Stand: 09.06.2017.
- [PB15] Price, T.; Barnes, T.: Comparing textual and block interfaces in a novice programming environment. In: Proceedings of the eleventh annual International Conference on International Computing Education Research. ACM, S. 91–99, 2015.
- [Re02] Resnick, M.: Rethinking learning in the digital age. The Global Information Technology Report: Readiness for the Networked World. Oxford University Press, 2002.
- [RMH14] Ruf, A.; Mühling, A.; Hubwieser, P.: Scratch vs. Karel: impact on learning outcomes and motivation. In: Proceedings of the 9th Workshop in Primary and Secondary Computing Education. ACM, S. 50–59, 2014.
- [RMMH09] Resnick, M.; Maloney, J.; Monroy-Hernández, A. et al.: Scratch: programming for all. Communications of the ACM, 52(11):60–67, 2009.
- [Sc17a] Scratch-Wiki: Cloning, wiki.scratch.mit.edu/wiki/Cloning, Stand: 09.06.2017.
- [Sc17b] Scratch-Wiki: Snap!, wiki.scratch.mit.edu/wiki/Snap, Stand: 09.06.2017.
- [St15] Strecker, K.: Grafische Programmiersprachen im Abitur. In (Jens Gallenbacher, Hrsg.): INFOS 2015: Informatik allgemeinbildend begreifen. Jgg. 16 in Lecture Notes in Informatics (LNI), Köllen, Bonn, S. 293–300, 2015.
- [SWM13] Squeak-Wiki: Morph, wiki.squeak.org/squeak/1820, Stand: 09.06.2017.
- [WW15] Weintrop, D.; Wilensky, U.: Using Commutative Assessments to Compare Conceptual Understanding in Blocks-based and Text-based Programs. In: ICER. Jgg. 15, S. 101–110, 2015.

Alltagsvorstellungen in der Informatik: Erhebungsmethodik und Implikationen für den Unterricht

Michael T. Rücker¹, Nils Pancratz², Carolin Gold-Veerkamp³, Niels Pinkwart⁴, Torsten Brinda⁵

Abstract: Ein zentrales Konzept jeder konstruktivistisch geprägten Auffassung von Lernen ist, dass das Vorwissen, die Präkonzepte und ggf. Fehlvorstellungen der Lernenden einen direkten Einfluss auf den Lernprozess haben: sowohl negativ als auch positiv. Speziell für die Informatik gilt, dass Lernende heutzutage von Beginn an in einer Welt aufwachsen, die von informatischen Artefakten und Systemen nahezu vollständig durchdrungen ist. Sie beobachten sie, interagieren mit ihnen und formen so Vorstellungen über ihre Funktionsweisen und Eigenschaften. Sie bilden somit bereits im Alltag und vor dem Beginn jeglicher Schulbildung kohärente Konzeptionen über zentrale Gegenstände und Inhalte der Informatik. Im Rahmen dieses Workshops werden zunächst verschiedene methodische Ansätze und erste Ergebnisse zu Erhebungen solcher Lernervorstellungen vorgestellt. Anschließend sollen diese anhand von drei Leitfragen verglichen und diskutiert werden: Welche Stärken und Schwächen haben die jeweils verwendeten empirischen Methoden bei der Erhebung von Lernervorstellungen in der Informatik? Wie können sie sich ggf. sinnvoll ergänzen? Welche Implikationen haben die erhobenen Vorstellungen für die Unterrichtspraxis?

Keywords: Konstruktivismus, Lernervorstellungen, Alltagsvorstellungen, Forschungsmethodik, Unterrichtspraxis

1 Alltagsvorstellungen in der Informatik

Informatische Artefakte, Produkte und Systeme sind heutzutage allgegenwärtig. Sie sind Teil der unmittelbaren sowie medialen Umwelt von Kindern und Jugendlichen. Sie beobachten sie, interagieren mit ihnen und formen so mentale Modelle und Konzeptionen darüber, wie diese Artefakte und Systeme funktionieren, welche Eigenschaften sie haben und wie sie ggf. zusammenhängen. Mit anderen Worten, sie formen bereits im Alltag kohärente Vorstellungen und Erklärungsmodelle für Dinge, die ihnen der Informatikunterricht vermitteln soll. Ein zentrales Konzept jeder konstruktivistisch geprägten Auffassung

¹ Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, rueckerm@informatik.hu-berlin.de

² Carl von Ossietzky Universität Oldenburg, Didaktik der Informatik, Uhlhornsweg 84, 26129 Oldenburg, nils.pancratz@uni-oldenburg.de

³ Hochschule Aschaffenburg, Fakultät Ingenieurwissenschaften, Würzburger Straße 45, 63743 Aschaffenburg, carolin.gold-veerkamp@h-ab.de

⁴ Humboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, pinkwart@informatik.hu-berlin.de

⁵ Universität Duisburg-Essen, Didaktik der Informatik, Schützenbahn 70, 45127 Essen, torsten.brinda@uni-due.de

von Lehren und Lernen ist, dass derartige Alltags- und ggf. Fehlvorstellungen einen signifikanten Einfluss auf den Lernprozess haben. Als „points of departure“ [Du07] bilden sie einerseits notwendige Voraussetzungen dafür, dass Lernen überhaupt stattfinden kann, können aber andererseits auch im Widerspruch zum Lernstoff stehen und den Lernprozess somit behindern. Es wird dann zwar gelernt, aber nicht unbedingt das, was beabsichtigt wurde.

Am ausführlichsten untersucht ist diese Thematik in der Didaktik der Naturwissenschaften. Duit hat hier eine ausführliche Bibliographie zusammengestellt [Du09]. Aber auch in der Informatik existiert ein zunehmendes Interesse und somit eine wachsende Literaturbasis zu Alltags- und Fehlvorstellungen [Be15, DWZ12, Kn11, Ra13, RP16, Se13]. Da es sich bei derartigen Vorstellungen um latente – d.h. nicht direkt beobachtbare – Konstrukte handelt, ist die Frage nach geeigneten Erhebungsmethoden von zentraler Bedeutung. Die Ansätze und somit auch die generierten Daten, die dazu in verschiedenen Arbeiten zum Einsatz kommen, rangieren von rein quantitativen und statistischen Verfahren [Be15], über Freitextfragebögen [Ra13] bis hin zu rein qualitativen und hoch interpretativen Interviewanalysen [Se13]. Die allgemeinen Stärken und Schwächen quantitativer und qualitativer Forschung sind hinreichend bekannt. Sie werden jedoch selten im konkreten Kontext ihrer jeweiligen Anwendung diskutiert. Letztlich schafft der konkrete Untersuchungsgegenstand weitere Anforderungen an ein Erhebungsinstrument. Angesichts dessen wird die Entscheidung für oder gegen eine bestimmte Methode leider nur selten explizit begründet oder gar im Vergleich zu möglichen Alternativen evaluiert. Welche Einsichten und Aussagen ermöglichen verschiedene methodische Ansätze überhaupt im konkreten Kontext der Erhebung von Lernervorstellungen? Sind einige Ansätze eventuell inhärent ungeeignet zur Erhebung derartiger latenter Konstrukte? Nicht zuletzt bleiben viele Arbeiten nach wie vor sehr vage bei der Frage, welche Konsequenzen ihre jeweils erhobenen Vorstellungen nun für die entsprechende Unterrichtspraxis haben.

Ziel dieses Workshops ist es, die Fragen nach geeigneten Erhebungsmethoden sowie der unterrichtlichen Relevanz von Lernervorstellungen in der Informatik weiterführend zu erörtern. Dazu sollen verschiedene methodische sowie inhaltliche Ansätze einander gegenübergestellt werden. Es soll herausgearbeitet werden, welche Vor- und Nachteile die jeweils angewandten Methoden im konkreten Kontext der Erhebung von Lerner-vorstellungen in der Informatik haben und wie sie sich ggf. komplementieren können. Nicht zuletzt soll diskutiert werden, welche Implikationen die jeweils erhobenen Vorstellungen nun für die Praxis des Informatikunterrichts oder -studiums haben.

2 Methodische Ansätze und erste Ergebnisse

Die im Folgenden vorgestellten Arbeiten wurden von den verschiedenen Co-Autor/innen dieses Beitrags bzw. innerhalb der jeweiligen Arbeitsgruppen an insgesamt vier verschiedenen deutschen Hochschulen unabhängig voneinander konzipiert und durchgeführt.

Ihnen liegen zwar einerseits sehr ähnliche Fragestellungen in Bezug auf Lernervorstellungen in der Informatik zugrunde. Sie verfolgen jedoch andererseits sehr unterschiedliche methodische Ansätze und ermöglichen daher verschiedene Perspektiven auf bzw. Einsichten in den Untersuchungsgegenstand.

2.1 Focus Interviews und Grounded Theory

Der erste Ansatz basiert auf der Theorie der konzeptuellen Kategorisierung [Mu04] und verwendet Focus Interviews und Methoden der Grounded Theory [Ch11] zur Erhebung von Schülervorstellungen zu informatischen Artefakten bzw. Geräten. Die zentrale Forschungsfrage ist: Welche technischen Geräte kategorisieren Schülerinnen und Schüler als „informatisch“ und wie gehen sie dabei vor?

Zur Beantwortung dieser Frage wurden 28 Schülerinnen und Schüler verschiedener Altersgruppen (Klasse 5 bis 13) und mit unterschiedlichem Wahlverhalten bzgl. des Informatikunterrichts individuell interviewt. In den Interviews wurden ihnen insgesamt 33 Bildkarten vorgelegt, auf denen verschiedene alltägliche und weniger alltägliche technische Geräte abgebildet waren, u.a. ein Taschenmesser, eine Plasmalampe, ein Toaster, ein Smartmeter, eine Waschmaschine, ein modernes Auto, ein Smartphone und der Curiosity Mars-Rover. Die Bilder waren nicht beschriftet. Den Proband/innen wurden folgende drei Aufgaben in unterschiedlicher Reihenfolge gestellt:

„Bitte lege all die Dinge zusammen in eine Gruppe, ...

- ...von denen du sagen würdest, dass sie mit *Informatik* zu tun haben oder dass ein Informatiker oder eine Informatikerin Ahnung davon hat oder sich damit beschäftigen würde.
- ...von denen du sagen würdest, dass sie eine Art *Computer* sind oder einen kleinen Computer enthalten.
- ...von denen du sagen würdest, dass sie *programmierbar* sind.“

Die Teilnehmenden wurden zudem angehalten, ihre Gruppierungen zu erläutern und zu begründen, warum sie welche Dinge zur Gruppe dazu gelegt haben oder nicht. Die Interviews wurden videographiert und anschließend transkribiert. Bei der Analyse und Interpretation der verschiedenen Gruppierungen und insbesondere der entsprechenden Argumentationsprozesse traten verschiedene Phänomene bzw. Vorstellungen hervor.

Für die Eingruppierung eines Artefaktes als *programmierbar* mussten in der Regel Eigenschaften wie Zeitmessung, Rechnen oder die Fähigkeit, auf bestimmte Bedingungen zu reagieren, vorhanden sein. Derartige Funktionen waren für die Lernenden häufig nicht unmittelbar transparent bzw. nachvollziehbar. Folglich musste irgendwas „dahinter stecken“ – eben ein Programm: „Ich weiß nicht, wie das sonst funktionieren soll.“ Simple Funktionen wie das Drehen eines Rotors oder das Erhitzen eines Drahtes waren leichter

nachvollziehbar: „Da sind keine großen Einstellungen dahinter. [...] Du drückst es runter und dann geht durch die Mechanik irgendwie die Wärme an.“

Für die Eingruppierung als *informatisch* oder *Computer* reichten aber auch die oben genannten Funktionen häufig nicht aus. Hier waren komplexere Fähigkeiten gefordert, wie Multifunktionalität, Adaptivität oder Netzwerkkommunikation. Die Funktionen einer Waschmaschine, die man nicht „umprogrammieren“ kann, „kann ein Steuerelektroniker machen, das muss jetzt kein Informatiker sein.“ Simple Programme, Schaltkreise, Platinen oder eingebettete Rechner wurden häufig als nicht mächtig genug wahrgenommen um als „echte“ Computer oder als informatisch angesehen zu werden.

2.2 Repertory Grid Interviews

Ein weiterer Ansatz befasst sich mit der Entwicklung, Durchführung und Analyse von *Repertory Grid Interviews*, um Schülervorstellungen von der Funktionsweise und dem Aufbau des Internets zu untersuchen. Basierend auf den allgemeinen Grundlagen von *Repertory Grid Interviews* [Fr95, Be85, Ke55] und inspiriert von den wenigen fachdidaktischen Untersuchungen, in denen die Methode bereits verwendet wurde [bspw. BLP03, Wi01, BJS98], befragten wir in einer ersten Untersuchung fünf 11- bis 13-jährige Schüler in Einzelinterviews. Dabei wurden den Versuchspersonen zwölf *Elemente* (*Internet, Computer, Server, Webseite, IP-Adresse, E-Mail, Kabel, Google, Daten, Router, (Daten-)Pakete* und *(Chat-)Programm*) vorgelegt, die zum einen das Grundgerüst an Begriffen ausmachen, die zu einer wissenschaftlichen Erklärung des Internets vonnöten sind, und zum anderen denjenigen Phänomenen und Artefakten entsprechen, die Jugendliche hauptsächlich mit dem Internet in Verbindung bringen. Zur Erhebung der *Konstrukte* wurden den Befragten iterativ je drei Elemente vorgelegt, mit der Aufforderung zu erörtern, inwiefern sich zwei dieser drei Elemente in einer Eigenschaft ähneln, in der sie sich vom dritten unterscheiden. Die Aussagen wurden auf Ton aufgezeichnet und in *Matrizen* (*Grids*), die zum Ende der Interviews von den Versuchsteilnehmern vervollständigt wurden, festgehalten. Im Zuge der Auswertung wurden zusätzlich Liniendiagramme erstellt, mit denen sich die Strukturen der Repertory Grids ohne nennenswerten Informationsverlust grafisch darstellen lassen und Ordnungsstrukturen und Zusammenhänge leichter zugänglich gemacht werden können [He09, BLP03].

Ein Vergleich mit Ergebnissen vorheriger Untersuchungen [DWZ12, Se13] zeigte, dass es mit Hilfe der durchgeführten Repertory Grid Interviews möglich war, typische Fehlvorstellungen und Misskonzepte vor allem von Servern, IP-Adressen und Google und zum Aufbau des Internets festzustellen. Dies zeigt, dass die Repertory Grid Methode durchaus eine interessante Alternative zu herkömmlichen Methoden zur Forschung nach Schülervorstellungen darstellt. Weitergehende Forschungen zur Analyse von kognitiven Ordnungsstrukturen und – im Speziellen – Teil-Ganzes-Beziehungen, die Lernerinnen und Lerner von informatischen Artefakten herstellen, mit Hilfe der Repertory Grid Methode sind zum Zeitpunkt der Niederschrift dieses Artikels in Planung.

2.3 Halbstrukturierter Online-Fragebogen

Aufgrund der Relevanz von Schlagworten wie „Big Data“ und „Datenschutz“ für die informatische Bildung in Schulen und deren Bezug zum Thema „Datenbanken“ wurden im Rahmen einer empirischen Untersuchung [Te16] im Jahr 2016 Schülervorstellungen von relationalen Datenbanken untersucht. Zur Erhebung wurde ein halbstrukturierter Online-Fragebogen eingesetzt, welcher zum einen Fragen zu Vorstellungen von Datenbanken und deren Anwendung, zum anderen Fragen zu informatischen Phänomenen mit indirektem Datenbankbezug enthielt. Im Einzelnen wurde gefragt:

1. Sicherlich hast du schon einmal etwas von „Datenbanken“ gehört. Wie stellst du dir so eine Datenbank eigentlich vor? Beschreibe deine Vorstellungen bitte in wenigen Sätzen.
2. Datenbanken speichern alle möglichen Arten von Daten. Welche Eigenschaften einer Datenbank hältst du für besonders wichtig?
3. Im Alltagsleben spielen Datenbanken eine wichtige Rolle. Nenne bitte einige Beispiele, wo dir in deinem täglichen Leben Datenbanken begegnen.
4. Im Folgenden werden dir sechs Situationen beschrieben, die du vielleicht selber schon erlebt hast. Erkläre bitte jeweils kurz, wie es dazu gekommen ist!

Exemplarisch werden zwei dieser Situationen angegeben:

- Du möchtest dir bei einem Onlinehändler (z.B. Amazon) eine neue Smartphone-Hülle kaufen. Unten auf der Seite siehst du die Mitteilung: „Kunden, die diese Hülle kauften, haben auch gekauft.“
- Du bist auf einem Ausflug in einer fremden Stadt unterwegs. Dein Smartphone zeigt dir an, wo es in deiner unmittelbaren Umgebung die nächste Eisdielen gibt.

Die Auswertung der Fragebögen erfolgte mittels qualitativer Inhaltsanalyse nach Mayring und lieferte in induktiver Vorgehensweise drei verschiedene Kategoriensysteme zu verschiedenen Aspekten der Betrachtung der Thematik. Besonders interessant ist das Kategoriensystem zur ersten Frage, welches die Struktur des ANSI/SPARC 3-Ebenen-Modell von Datenbankmanagementsystemen widerspiegelt. Dabei werden interne, konzeptuelle und externe Aspekte unterschieden und zwar einmal aus informatisch-fachlicher Sicht und einmal aus Sicht deren lebensweltlicher Entsprechung.

2.4 Halbstrukturierte Interviews

Gemäß den Ergebnissen der jährlichen JIM-Studien (www.mpfs.de) kann im Jahr 2017 von einer nahezu flächendeckenden Verbreitung und recht intensiven, täglichen Nutzung von Smartphones durch Jugendliche ausgegangen werden. Smartphones werden schon seit einiger Zeit in der informatischen Bildung sowohl als Medium, als auch als Unterrichtsgegenstand betrachtet. Entwicklungen wie der MIT App Inventor nutzen die große Popularität für die Initiierung von Informatikinteresse aus. Für entsprechenden Unterricht nach dem Modell der didaktischen Rekonstruktion ist es deshalb wünschenswert, genauere

Kenntnisse über Schülervorstellungen in diesem Bereich zu erhalten. In einer empirischen Untersuchung [Br16] wurde deshalb eine explorative Studie zu Schülervorstellungen von Smartphones durchgeführt mit dem Ziel, einen ersten Überblick über bestehende Vorstellungen zu erhalten. Dazu fanden zwölf halb-strukturierte Interviews mit einer Reihe von Gymnasialschülern statt. Inhaltliche Schwerpunkte der Interviews waren:

- drahtlose Netzverbindung (Netzwerkstruktur von Mobilfunknetzen, Adressierung in Mobilfunknetzen, Übertragungsmedium, Handover, Internetprotokoll)
- Apps (Definition, Aufruf, Herkunft, Installation)
- Steuerung mittels Touchscreens (Bedienelemente, Touchscreen, Programmarchitektur)
- kompakte Bauart (Kompaktheit der Hardware, technische Grenzen)

Acht der Interviews wurden ausgewählt, transkribiert und einer strukturierenden, qualitativen Inhaltsanalyse nach Mayring unterzogen. Die Kategorienbildung erfolgte deduktiv, basierend auf den Schwerpunkten des Interviewleitfadens. Innerhalb der Kategorien wurden inhaltlich zusammengehörige Vorstellungen zusammengefasst.

2.5 Spezielle Items

Basierend auf einer Arbeit zur Untersuchung von Kategorisierungsstrategien in der Biologie wurde eine empirische Studie durchgeführt, in welcher Schülerinnen und Schüler verschiedener Altersgruppen dazu aufgefordert wurden, rund 20 Begriffe aus der Informatik (darunter: Computer, Internet, Roboter, Bildschirm, YouTube, Festplatte etc.) durch verschiedenfarbige Unterstreichung Kategorien zuzuordnen und diese Kategorien dann anschließend selbst zu benennen. Vorschläge für fünf Farben und damit fünf Kategorien waren angegeben, ferner Platz für eine weitere Kategorie. Darüber hinaus wurden die Teilnehmenden gefragt, welche Begriffe ihrer Meinung nach in der Liste fehlten und welche ihrer Meinung nach nicht kategorisierbar waren.

Insgesamt wurden über 400 Fragebögen erfasst und die erstellten Kategorien im Hinblick auf Begriffszuordnungen sowie erkennbare Kategorisierungsstrategien analysiert. Eine weiterführende Analyse mittels Clusteranalyse ist zum Zeitpunkt des Verfassens dieses Textes in Vorbereitung.

2.6 Pretest und Grounded Theory

Betrachtet man die Hochschulbildung, könnte man vermuten, dass Lernende ein differenzierteres Bild der Informatik besitzen, da sie sich – vermeintlich – bewusst für ein informatiknahes Studium entschieden haben. In diesen Studiengängen besitzt v. a. das Software Engineering (SE) einen hohen Stellenwert.

Das zunehmende wissenschaftliche Interesse an Schülervorstellungen in der Informatik

(vgl. Kap. 1) lässt sich kaum auf SE übertragen [Su10, Iv06]. Daher gilt es, eine explorative Grundlagenforschung basierend auf der Grounded Theory [Ch11] mit der Frage „Welche Vorstellungen haben Studierende zu Software Engineering?“ zu begründen.

In einem Pretest wurden Erstsemester⁶ (N = 217) implizit gebeten zu unterscheiden: „Bitte beschreiben Sie die nachfolgenden drei Begriffe (Informatik, Programmieren und SE) so genau wie möglich mit Ihren eigenen Worten.“ Die handschriftlichen Freitexte wurden im Nachhinein digitalisiert und offen kodiert [Ch11].

Bei der anschließenden Analyse fällt auf, dass...

- ... quantitativ gesehen, beim Begriff „SE“ die wenigsten Antworten – unabhängig vom Studienfach – gegeben wurden. So haben 18 %⁷ das Freitextfeld leer gelassen.
- ... der Terminus Informatik mit dem *Computer/Rechner/PC*, der Tätigkeit des *Programmierens* und mit der *Daten-/Informationsverarbeitung* assoziiert wurde.
- ... der Begriff Informatik umschrieben werden kann als „*Allgemein alles, was mit Software zu tun hat*“ oder „*Alles, was mit Computern zu tun hat*“.
- ... die Beschreibung der Studierenden zu SE zeigt, dass damit ein *Entwicklungsprozess* gemeint ist, dessen Ergebnis eine *Software* ist, die *programmiert* wird. Vergleicht man dies mit der Erklärung zum Programmieren, fallen kaum Unterschiede auf: Hier entsteht ebenfalls eine *Software* durch die Tätigkeit des *Schreibens* mittels einer *Programmiersprache*.⁸ Zudem wird hier ebenfalls eine *Entwicklungstätigkeit* einbezogen.

Es lassen sich demnach einige Schlussfolgerungen ziehen:

- Es scheint, dass viele Studierende keine konkrete Vorstellung von SE haben.
- Außerdem scheinen die Studierenden, die über ein Bild von SE verfügen, keine großen Unterschiede zum Programmieren aufzeigen zu können.
- Weiterhin könnte man inferieren, dass Informatik für viele Studierende ein Oberbegriff für Computer, Software etc. darstellt.

Um validere Aussagen treffen zu können, sollten weitere Gruppen befragt werden, z. B. Studierende der Informatik/Ingenieurwissenschaften anderer Hochschulen. Somit wäre es möglich die bisherigen Daten in Form einer Querschnittstudie auszuweiten.

⁶ Studiengänge: Mechatronik (N = 80), Elektro- und Informationstechnik (N = 48) und Informatik (N = 89).

⁷ Im Vergleich dazu haben 100 % der Befragten den Begriff Programmieren beschrieben und nur 6 % keine Antworten zum Fachwort Informatik gegeben.

⁸ Umfasst sowohl Nennungen des Überbegriffs wie auch einzelner konkreter Sprachen (z. B. C, C++, Java).

3 Aufbau des Workshops

Im Rahmen des Workshops sollen die oben angeführten Arbeiten zunächst in Form von etwa 10-minütigen Kurzvorträgen etwas genauer vorgestellt und vor allem auch mittels konkretem Datenmaterial illustriert werden. Der Workshop ist zudem offen für weitere Beiträge von anderen Teilnehmerinnen und Teilnehmern, die ihre eigenen Ansätze oder auch eigene Erfahrungen aus der Praxis präsentieren und zur Diskussion stellen wollen.

Im Anschluss sollen alle präsentierten Beiträge hinsichtlich der bereits genannten Leitfragen verglichen und diskutiert werden:

1. Was sind allgemeine Stärken und Schwächen der verschiedenen Ansätze hinsichtlich der Erhebung von Lernervorstellungen in der Informatik?
2. Wie unterscheiden sich die Ansätze hinsichtlich der Qualität der Aussagen, die auf ihrer Basis über Lernervorstellungen möglich sind und wie können sie sich gegebenenfalls ergänzen?
3. Welche Implikationen haben die vorgestellten Lernervorstellungen für die Praxis des Informatikunterrichts bzw. -studiums?

Literaturverzeichnis

- [Be85] Beail, Nigel: Repertory Grid Technique and personal constructs: applications in clinical & educational settings. Croom Helm, London [et.al.], 1985.
- [Be15] Bergner, Nadine: Konzeption eines Informatik-Schülerlabors und Erforschung dessen Effekte auf das Bild der Informatik bei Kindern und Jugendlichen. Dissertation, RWTH Aachen, 2015.
- [BJS98] Baxter, Irene A.; Jack, Frances R.; Schröder, Monika J.A.: The use of Repertory Grid Method to elicit perceptual data from primary school children. Food Quality and Preference, 9(1 - 2): S. 73–80, 1998.
- [BLP03] Bruder, Regina; Lengnink, Katja; Prediger, Susanne: Wie denken Lehramtsstudierende über Mathematikaufgaben? Ein methodischer Ansatz zur Erfassung subjektiver Theorien mittels Repertory-Grid-Technik. Mathematica Didactica, 1(26): S. 63–85, 2003.
- [Br16] Braun, Friederike Victoria: Schülervorstellungen von Smartphones. Schriftliche Hausarbeit im Rahmen der Ersten Staatsprüfung für das Lehramt an Gymnasien/Gesamtschulen. Universität Duisburg-Essen, 2016.
- [Ch11] Charmaz, Kathy: Constructing Grounded Theory. SAGE Publication, 1. Aufl., 2011.
- [Du07] Duit, Reinders: Science education research internationally: Conceptions, research methods, domains of research. Eurasia Journal of Mathematics, Science & Technology Education, 3(1), 3-15, 2007

- [Du09] Duit, Reinders: Bibliography: Students' and Teachers' Conceptions and Science Education, 3 2009. Last accessed: 07.11.2016.
- [DWZ12] Diethelm, Ira; Wilken, Henning; Zumbrägel, Stefan: An investigation of secondary school students' conceptions on how the Internet works. In: Proceedings of the 12th Koli Calling International Conference on Computing Education Research. ACM, S. 67–73, 2012.
- [Fr95] Fromm, Martin: Repertory Grid Methodik: ein Lehrbuch. Weinheim: Dt.-Studien-Verl., 1995.
- [He09] Helmerich, Markus Alexander: Liniendiagramme in der Wissenskommunikation - Eine mathematisch-didaktische Untersuchung. Dissertation, Technische Universität, Darmstadt, 2009.
- [Iv06] Ivins, Jim; von Kinsky, Brian R.; Cooper, David; Robey, Michael: "Software Engineers and Engineering: A Survey of Undergraduate Preconceptions" In: Frontiers in Education. 36th Annual Conference, S. 6-11, 2006.
- [Ke55] Kelly, George A.: The psychology of personal constructs. Norton, New York, 1955.
- [Kn11] Knobelsdorf, Maria: Biographische Lern- und Bildungsprozesse im Handlungskontext der Computernutzung. Dissertation, FU Berlin, 2011.
- [Mu04] Murphy, Gregory L.: The Big Book of Concepts. MIT Press, 2004.
- [Ra13] Rabel, Magnus: Grundvorstellungen zu Klassen und Objekten. In: Breier, Norbert; Stechert, Peer; Wilke, Thomas: Informatik erweitert Horizonte. Köllen, Bonn, S. 57–66, 2013.
- [RP16] Rucker, Michael T.; Pinkwart, Niels: Review and Discussion of Children's Conceptions of Computers. Journal of Science Education and Technology, 25(2), 2016.
- [Se13] Seifert, Oliver; Sauck, Tony; Schwarzbach, Maximilian; Lerch, Christopher; Weinert, Martin; Knobelsdorf, Maria: "Ich glaube, Google ist so was wie eine Vorhalle des Internets" – Erste Ergebnisse einer qualitativen Untersuchung von Schüler-vorstellungen von der Suchmaschine Google. In: Breier, Norbert; Stechert, Peer; Wilke, Thomas: Informatik erweitert Horizonte. Köllen, Bonn, S. 45–56, 2013.
- [Su10] Sudol, Leigh Ann; Jaspán, Ciera: "Analyzing the Strength of Undergraduate Misconceptions About Software Engineering" In: Sixth International Workshop on Computing Education Research, ser. ICER '10. New York, NY, USA: ACM, S. 31–40, 2010.
- [Te16] Terjung, Torsten: Schülervorstellungen von relationalen datenbanken: Eine empirische Untersuchung. Masterarbeit. Universität Duisburg-Essen, Fakultät für Wirtschafts-wissenschaften, 2016.
- [Wi01] Williams, Steven R.: Predications of the Limit Concept: An Application of Repertory Grids. Journal for Research in Mathematics Education, 32(4): S. 341–367, 2001.

Poster

Blended-Learning-Module für ein Lehrerfortbildungskonzept zum Informatikunterricht

Kensuke Akao¹

Abstract: Der Fortbildungsbedarf für Informatiklehrkräfte ist als hoch einzuschätzen, doch sehen viele Lehrkräfte Hürden, um an Fortbildungen teilzunehmen. Zwar ist die Flexibilität bezüglich Zeit und Ort eine Stärke von E-Learning, aber es ist noch ungeklärt, ob E-Learning einen wesentlichen Beitrag zur grundständigen Ausbildung von Informatiklehrkräften sein kann. Wir untersuchen das Potential von Blended-Learning für die Fortbildung von Informatiklehrkräften unter Verwendung von Prototyp-Modulen. Ziel ist die Entwicklung eines Lern-/Lehrdesigns für den Einsatz von E-Learning in Blended-Learning Szenarien, um höhere Teilnahmechancen an einer Informatik-Lehrerfortbildung zu schaffen.

Keywords: Lehrerfortbildung, Informatikunterricht, E-Learning, Blended-Learning, Mediendidaktik, Backup, Datenschutz

1 Chancen und Probleme von E-Learning für Fortbildungen

In Nordrhein-Westfalen (NRW) wurden viele Informatiklehrkräfte nachqualifiziert, besonders in der Sekundarstufe I (Sek-I). Oft reicht das Fachwissen nachqualifizierter Lehrkräfte für einen anspruchsvollen und professionellen Informatikunterricht, der wirklich den Vorgaben und Zielen des Kernlehrplans gerecht wird, nicht aus. Zusätzlich fällt es Lehrkräften oft aus organisatorischen Gründen schwer an Lehrerfortbildungen teilzunehmen, die ihren Unterricht verbessern könnten.

Die zeitliche und örtliche Flexibilität von E-Learning ist ein häufiger Vorschlag zur Lösung dieses Problems. *Blended-Learning* (Online-Module mit Präsenzterminen) bewahrt auch einen *Face to Face* Austausch [Ke13]. Einige Projekte, z. B. VLIN in Niedersachsen [VL17] oder FLIEG in Bayern [Sp09], haben Informatiklehrkräfte mithilfe von Blended-Learning ausgebildet. Ein Ergebnis unserer Umfrage² ist, dass sich die meisten Informatiklehrkräfte in NRW eine Fortbildung innerhalb der Dienstzeit wünschen (nur 4,5% können sich eine Fortbildung am Wochenende vorstellen und nur 1,2% am Abend). D. h. die Flexibilität von E-Learning oder Blended-Learning als 24 Std.-Lernumgebung scheint durch die Bereitschaft deutscher Lehrkräfte begrenzt.

¹ Westfälische Wilhelms-Universität(WWU) Münster, Didaktik der Informatik(DDI), Fliednerstrasse 21, 48149 Münster, kensuke.akao@uni-muenster.de

² Online-Umfrage vom 5.Sep.2016 bis 9.Feb. 2017 in NRW zum Informatikangebot Sek-I und zum Fortbildungsbedarf: 247 Rückmeldungen (202 Informatiklehrkräfte; 25 Lehrkräfte, die Fächer mit informatischen Inhalte unterrichten; 20 stellvertretende- / Schulleiter)

2 Untersuchungsstrategie des erwogenen Fortbildungskonzepts

Wir stellen uns die Forschungsfrage, ob Blended-Learning im Bereich der Lehrerfortbildung für den Informatikunterricht wirkungsvoll eingesetzt werden kann. Zur Untersuchung der Akzeptanz von Blended-Learning erstellen wir zunächst ein neues Lehrerfortbildungsprojekt mit Prototyp-Modulen das Thema „Datensicherheit, Datenschutz und Datenbank“. Und das erste Lernmodul ist „Backup für Daten und Systeme“. Ziel ist die Schaffung höherer Teilnahmekancen für Lehrerfortbildungen, indem (a) die Anzahl der Präsenztermine möglichst verringert wird und (b) die benötigte Lernzeit durch eine effektive Präsentation der Lerninhalte verkürzt wird.

Die Fortbildungen bestehen aus Online-Modulen (Theorie-Phase), gefolgt von einer Praxis-Veranstaltung (Anwendungs-Phase). Für die Online-Module wurden Videos mit animierten Grafiken entwickelt, um komplizierte Inhalte möglichst kompakt darzustellen. Um die Akzeptanz verschiedener Medien vergleichen zu können, bieten wir die Inhalte auch als reine Audioversion oder reines Textdokument mit Grafiken an. Nach jeder Phase werden die Teilnehmer über das Konzept befragt, damit die Fortbildung in einem Verbesserungszyklus *Design-Based-Research* [DRC03] stetig weiterentwickelt werden kann.

3 Ausblick

Aktuell planen wir zu untersuchen, inwiefern informelle Austauschmöglichkeiten die Anzahl der formellen Präsenztermine weiter reduzieren können. Beispielweise waren 40,9% der Lehrkräfte bei unserer Umfrage grundsätzlich an einem regelmäßigen Stammtisch interessiert. In den nächsten Monaten startet der erste Probelauf unseres Fortbildungskonzeptes. Wir arbeiten weiter daran, eine gehaltvolle, gern und gut besuchte Lehrerfortbildung anzubieten, welche gut auf die Bedürfnisse der Lehrkräfte abgestimmt werden kann.

Literaturverzeichnis

- [DRC03] The Design-Based Research Collective: Design-Based Research: An Emerging Paradigm for Educational Inquiry, *Educational Researcher* Vol.32 No.1, S.5-8, 2003.
- [Ke13] Kerres, M.: *Mediendidaktik – Konzeption und Entwicklung mediengestützter Lernangebote* 4.Auflage, 2013.
- [Sp09] Spohrer, M.: *Konzeption und Analyse neuer Maßnahmen in der Fort- und Weiterbildung von Informatiklehrkräften*, Dissertation an der Technische Universität München, 2009
- [VL17] *Virtuellen Lehrerweiterbildung Informatik in Niedersachsen (VLIN)*, vlin.de/ Stand: 5.1.2017.

Unterschiede beim Memorieren von Quelltexten zwischen NovizInnen und ExpertInnen der objektorientierten Programmierung

Mike Barkmin¹, Matthias Kramer², David Tobinski³, Torsten Brinda⁴

Abstract: Dieser Artikel befasst sich mit den unterschiedlichen Vorgehensweisen von ExpertInnen und NovizInnen der Programmierung beim Memorieren und Interpretieren von Quelltexten. ExpertInnen haben die Quelltexte überwiegend auf einer semantischen Ebenen memoriert, NovizInnen hingegen zeilenweise.

Keywords: Objektorientierte Programmierung, Arbeitsgedächtnis, Programmiererfahrung.

1 Einleitung

NovizInnen der Programmierung stehen oft vor dem Problem, mehrere Herausforderungen bewältigen zu müssen, wenn sie eine Programmieraufgabe lösen. Zum einen müssen sie einen Problemlöseprozess durchlaufen und passende Algorithmen entwickeln, zum anderen müssen sie die syntaktischen Elemente der verwendeten Programmiersprache beherrschen und richtig anwenden. Im Rahmen einer im Kontext des COMMOOP-Projekts [KHB16] durchgeführten Untersuchung wurden frühere Ergebnisse bestätigt, nach denen die Beherrschung syntaktischer Elemente für NovizInnen eine Herausforderung darstellt. NovizInnen orientieren sich eher an gelernten Quelltextkonventionen als an der Semantik. Basierend auf diesen Ergebnissen und denen von Adelson [Ad81] wurde postuliert, dass ExpertInnen Quelltexte in semantischen Einheiten im Arbeitsgedächtnis ablegen, wohingegen NovizInnen syntaktische Einheiten verwenden. Um diese Hypothese zu überprüfen, wurde ein Test zum Memorieren von Quelltexten entwickelt.

2 Methode, Ergebnisse und Diskussion

Die ProbandInnen wurden gebeten, verdeckte Quelltexte und natürlichsprachliche Texte, welche nach Belieben aufgedeckt werden konnten, in ein Textfeld zu übertragen. Während

¹ Univ. Duisburg-Essen, DDI, Schützenbahn 70, 45127 Essen, Deutschland mike.barkmin@stud.uni-due.de

² Univ. Duisburg-Essen, DDI, Schützenbahn 70, 45127 Essen, Deutschland matthias.kramer@uni-due.de

³ Univ. Duisburg-Essen, Institut für Psychologie, Universitätsstr.2, 45141 Essen, Deutschland david.tobinski@uni-due.de

⁴ Univ. Duisburg-Essen, DDI, Schützenbahn 70, 45127 Essen, Deutschland torsten.brinda@uni-due.de

des Vorgangs wurden alle Schritte protokolliert. Zusätzlich wurden drei psychologische Tests durchgeführt, um die allgemeine Merkfähigkeit der ProbandInnen kontrollieren zu können. Anschließend sollten die ProbandInnen eine Selbsteinschätzung ihrer Programmiererfahrung anhand validierter Likert-Skalen vornehmen (siehe [Ba17]). An der Studie haben 42 Studierende (30: Informatik, 12: andere) teilgenommen. Die protokollierten Tastenanschläge und Anschauphasen wurden genutzt, um Unterschiede zwischen NovizInnen und ExpertInnen der objektorientierten Programmierung zu ermitteln. Die Überprüfung eines aus der Theorie abgeleiteten Strukturgleichungsmodells (siehe Abb. 1 links) mittels PLSPM hat gezeigt, dass die Programmiererfahrung (PE) die Fähigkeit in den Übertragungsaufgaben (F) am besten erklärt. Das Arbeitsgedächtnis (AG) und die Konzentrationsfähigkeit (K) sind diesbezüglich schwächere Indikatoren. Eine Bootstrapping-Validierung ergab, dass nur die Pfade von PE nach F und von K nach AG signifikant sind. Damit sind quantitative Hinweise gefunden, dass mit mehr Programmiererfahrung wahrscheinlich mehr Informationseinheiten eines Quelltexts memoriert werden können. Dies ist wahrscheinlich auf das Zusammenfassen von Informationseinheiten zurückzuführen. Des Weiteren wurde untersucht, ob die

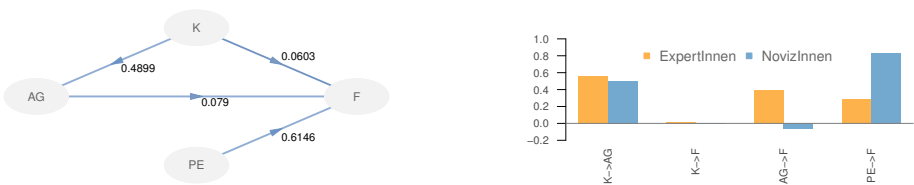


Abb. 1: Strukturmodell und Gruppenunterschiede der Pfadkoeffizienten

Pfadkoeffizienten sich zwischen NovizInnen und ExpertInnen (operationalisiert durch die PE) unterscheiden. Die unterschiedlichen Pfadkoeffizienten (siehe Abb. 1 rechts) sind nur für den Pfad PE nach F signifikant. Daher ist der Einfluss der Programmiererfahrung bei NovizInnen deutlich größer als bei ExpertInnen.

Literatur

- [Ad81] Adelson, B.: Problem solving and the development of abstract categories in programming languages. *Memory & Cognition* 9/4, S. 422–433, 1981.
- [Ba17] Barkmin, M.: Konstruktion und Erprobung eines Bausteins zur Kompetenzmessung im Bereich der objektorientierten Programmierung in den Dimensionen Syntax und Semantik. Universität Duisburg-Essen, 2017.
- [KHB16] Kramer, M.; Hubwieser, P.; Brinda, T.: A Competency Structure Model of Object-Oriented Programming. In: 2016 International Conference on Learning and Teaching in Computing and Engineering (LaTICE). S. 1–8, März 2016.

Wirkmodelle zum Computational Thinking in der Grundschule

Christine Bescherer¹ und Andreas Fest¹

Abstract: Im Rahmen des von der Telekom Stiftung geförderten Projekts „Digitales Lernen in der Grundschule Stuttgart/Ludwigsburg“ werden in Seminaren Unterrichtseinheiten zur Förderung des „Computational Thinking“ entwickelt und von den Studierenden in einer Grundschule erprobt. Im Verlauf des Projekts wird über kontinuierlich verfeinerte Wirkmodelle die Wirkung dieser Interventionen sowohl auf Ebene der Studierenden wie auch der Schülerinnen und Schüler erfasst und dargestellt.

Keywords: Computational Thinking in der Grundschule, Wirkmodelle, Interventionsprojekt

1 Einleitung und theoretischer Hintergrund

Die Entwicklung digitaler Lehr-/Lernszenarien in der Grundschule ist das Ziel des von der Telekom Stiftung geförderten Projekts „Digitales Lernen in der Grundschule Stuttgart/Ludwigsburg“ (dileg-SL, <https://www.ph-ludwigsburg.de/16553.html>) an der Pädagogischen Hochschule Ludwigsburg. Neben dem produktiven und kritischen Umgang mit digitalen Medien in verschiedenen Kontexten ist eines der wichtigen Ziele die Unterstützung der Grundschülerinnen und Grundschüler bei der Entwicklung informatischer und algorithmischer Grundkompetenzen. Dazu entwickeln Studierende des Lehramts Grundschule unter Anleitung Lehr-/Lernszenarien („halb-gare Mikrowelten“ [Pa75], [Ky07]), die das „Computational Thinking“ [Gr13] von Schülerinnen und Schülern im Mathematikunterricht der 3. oder 4. Klasse unterstützen.

2 Methode und Durchführung

In dem Projekt werden Lehramtsstudierende (Lehramt Grundschule) in einem Seminar nach dem Kennenlernen und eigenen Erfahrungen zum „Computational Thinking“ unter Anleitung „halb-gare Mikrowelten“ [Ky07] zu mathematischen Themen der Grundschule entwickeln, in denen die Schülerinnen und Schüler verschiedene Elemente des „Computational Thinking“ erlernen, erleben und reflektieren sollen. Diese Mikrowelten werden dann in einer Grundschule in passenden Lernsettings (Projektunterricht, Regelunterricht, Medien-AG) erprobt, videographiert und später im Seminar wieder

¹ Pädagogische Hochschule Ludwigsburg, Institut für Mathematik und Informatik, Reuteallee 46, 71634 Ludwigsburg, bescherer@ph-ludwigsburg.de, fest@ph-ludwigsburg.de

reflektiert.

Die erwartete Wirkung bezieht sich v.a. auf den Zuwachs an Fähigkeiten im Umgang mit den Elementen des Computational Thinkings sowie z.B. auch Änderungen in der computerspezifischen Selbstwirksamkeitserwartung. Da durch die geringen Fallzahlen sowohl der Studierenden als auch bei den Schülerinnen und Schülern im Rahmen einer solchen Entwicklungs- und Machbarkeitsstudie kein experimentelles Design zum Nachweis der Wirksamkeit leistbar ist, wird die Wirkung anhand der iterativen Verfeinerung von Wirkmodellen (vgl. [Wa16]) dokumentiert.

Das Ausgangsmodell umfasst den Input, die Maßnahme und den erwarteten Output noch in einer undifferenzierten Form:

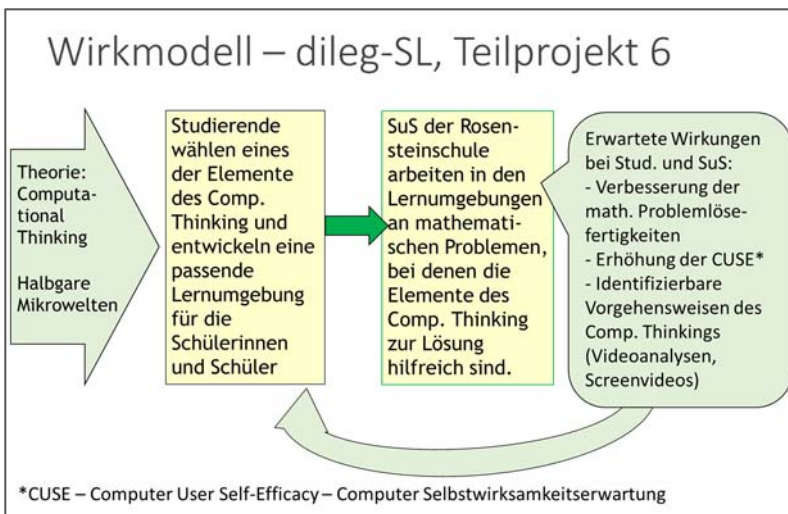


Abb. 1: Das undifferenzierte Wirkmodell in seiner Ausgangsform.

Im Verlauf des Projekts wird das Wirkmodell immer weiter verfeinert und sowohl der theoretische Input, die verwendeten Lernszenarien als auch die Instrumente zu Erfassung der Änderungen werden nach und nach konkretisiert.

Literaturverzeichnis

- [Gr13] Grover, S.; Pea, R.: Computational Thinking in K–12 A Review of the State of the Field. In Educational Researcher, 42. Jg., Nr. 1, S. 38-43, 2013.
- [Ky07] Kynigos, C.: Half-baked logo microworlds as boundary objects in integrated design. In Informatics in Education - An International Journal, Vol 6_2, S. 335-359. 2007.
- [Wa16] Wachsmuth, E. & Hense, J.: Wirkmodelle zur Unterstützung der Evaluation komplexer Hochschulprojekte. In: Qualität in der Wissenschaft. 3+4/2016. Universitäts-Verlag Weblar. Bielefeld. S. 80-87, 2016.

Schülervorstellungen von relationalen Datenbanken: eine empirische Untersuchung

Torsten Brinda¹ und Thorsten Terjung²

Abstract: Als Beitrag zur didaktischen Rekonstruktion verknüpft die vorliegende empirische Untersuchung individuelle Schülervorstellungen zum Thema relationale Datenbanken mit den zugehörigen informatischen Fachkonzepten. Zur Erhebung wurde ein Online-Fragebogen eingesetzt, welcher zum einen Fragen zu Vorstellungen von Datenbanken und deren Anwendung, zum anderen Fragen zu informatischen Phänomenen mit indirektem Datenbankbezug enthielt. Die Auswertung der Fragebögen erfolgte mittels qualitativer Inhaltsanalyse nach Mayring und lieferte drei Kategoriensysteme, wobei die Kategorisierung der Schülervorstellungen analog zum ANSI/SPARC 3-Ebenen-Modell als wichtigstes Ergebnis zu nennen ist. Dieses Modell konnte dabei in zwei unterschiedlichen Ausprägungen in den Schülervorstellungen identifiziert werden – einerseits als Betrachtung von informatischen Datenbankkonzepten, andererseits mit Blick auf die realweltliche, analoge Speicherung und Verarbeitung von Daten. Schlussfolgernd kann für die didaktische Strukturierung des Unterrichts empfohlen werden, die verschiedenen Ebenen des ANSI/SPARC-Modells von Beginn an zu thematisieren und deutlich voneinander zu unterscheiden.

Keywords: Schülervorstellungen, Datenbanken, empirische Studie, Kategorisierung.

1 Einleitung

Als Umsetzung des Modells der didaktischen Rekonstruktion (vgl. [Di11]) verknüpft die empirische Untersuchung individuelle Schülervorstellungen mit informatischen Fachkonzepten zum Thema relationale Datenbanken. Dabei wird beides im wechselseitigen Vergleich auf Gemeinsamkeiten und Unterschiede untersucht und daraus eine didaktische Strukturierung für den Unterricht abgeleitet. Zur Erhebung der Schülervorstellungen wurde ein Online-Fragebogen eingesetzt, welcher zum einen Fragen zu Vorstellungen von Datenbanken und deren Anwendung, zum anderen Fragen zu informatischen Phänomenen mit indirektem Datenbankbezug enthält.

2 Ergebnisse und Fazit

Die Auswertung der Fragebögen erfolgte mittels der qualitativen Inhaltsanalyse nach

¹ Didaktik der Informatik, Universität Duisburg-Essen, Schützenbahn 70, 45127 Essen, torsten.brinda@uni-due.de

² Institut für Informatik u. Wirtschaftsinformatik, Universität Duisburg-Essen, Schützenbahn 70, 45127 Essen, thorsten.terjung@stud.uni-due.de

Mayring [Ma10] und lieferte drei verschiedene Kategoriensysteme zu verschiedenen Aspekten der Betrachtung der Thematik. Zentral ist hier zunächst die Kategorisierung der Schülervorstellungen analog zum ANSI/SPARC 3-Ebenen-Modell [TS78], welches bei Datenbankmanagementsystemen die interne Ebene der physischen Speicherung der Daten, die konzeptuelle Ebene der Strukturierung der Daten und die externe Ebene der benutzerdefinierten Sicht auf die Daten unterscheidet. Dieses Modell konnte in zwei Ausprägungen in den Schülervorstellungen identifiziert werden. Einerseits als Betrachtung von digitalen Datenbankkonzepten, andererseits mit Blick auf die realweltliche, analoge Speicherung und Verarbeitung von Daten. Weiterhin zeigte sich, dass bei Schülerinnen und Schülern mit wenig bis keiner unterrichtlichen Erfahrung in der Datenbankthematik häufig eine Vorstellung auf der internen Ebene vorherrscht, während mit ansteigender Behandlung des Themas im Informatikunterricht ein Wechsel zur konzeptuellen Ebene erkennbar ist. Auch bei den genannten lebensweltlichen Begegnungen mit Datenbanken zeigte sich eine Abhängigkeit von der Thematisierung im Unterricht. Während Anfänger eher abstraktere Facetten wie Smartphone oder Laptop benennen, rücken bei den Schülerinnen und Schülern, die das Thema ausführlicher behandelt hatten, persönlich erfahrene Anwendungen wie etwa beim Online-Einkauf oder im Schulzusammenhang in den Fokus. Die Erklärungsansätze der Schülerinnen und Schüler zu den vorgestellten Phänomenen offenbaren einen teils deutlichen Mangel an Wissen bezüglich der tatsächlichen Anwendungen von Datenbanken und der zugrundeliegenden Fachkonzepte. Insbesondere die Auswertung und Analyse von Nutzerprofilen (Data Warehouse- und Data Mining-Konzept) ist den Schülerinnen und Schülern nur wenig bekannt. Die Verbindung mehrerer Teilaspekte bei der Auswertung verschiedener Datenbanken konnte nicht zufriedenstellend erklärt werden. Schlussfolgernd kann für die didaktische Strukturierung des Unterrichts empfohlen werden, die verschiedenen Ebenen des ANSI/SPARC-Modells von Beginn an zu thematisieren und deutlich voneinander zu unterscheiden. Der Schwerpunkt sollte dabei auf der Strukturierung und Ordnung von Daten liegen, wobei zur konkreten Umsetzung im Informatikunterricht lebensweltliche Phänomene mit Bezug zu Internetanwendung als geeignet erscheinen. Eine ausführliche Darstellung der Studie findet sich in [Te16].

Literaturverzeichnis

- [Di11] Diethelm, I., Dörge, C., Mesaros, A. M., & Dünnebier, M.: Die Didaktische Rekonstruktion für den Informatikunterricht. In: Thomas, M. (Hrsg.): Informatik in Bildung und Beruf. Köllen, Bonn, S. 77-86, 2011.
- [Ma10] Mayring, P.: Qualitative Inhaltsanalyse. Grundlagen und Techniken. 11. Auflage. Beltz, Weinheim, 2010.
- [Te16] Terjung, T.: Schülervorstellungen von relationalen Datenbanken. Masterarbeit. Universität Duisburg-Essen, Fakultät für Wirtschaftswissenschaften, 2016.
- [TK78] Tsichritzis, D.; Klug, A.: The ANSI/X3/SPARC DBMS framework report of the study group on database management systems. In: Inform. Systems 3 (3), S.173-191, 1978.

Die Handykiste für die Grundschule

Wibke Duwe¹

Abstract: Auf Grundlage der Forderungen der Kultusministerkonferenz zur digitalen Bildung in der Grundschule wird vorliegend ein Unterrichtskonzept zum Thema „Handy in der Grundschule“ vorgestellt. Da Mobiltelefone als komplexe IT-Systeme zu verstehen sind, können anhand dieser digitalen Grundkompetenzen, wie sie auch von der KMK gefordert werden, vermittelt werden.

Keywords: Digitale Bildung, Grundschule, Sachunterricht, Handy.

1 Einleitung

Mit der im Dezember 2016 beschlossenen Strategie „Bildung in der Digitalen Welt“ legt die Kultusministerkonferenz (KMK) ein neues Handlungskonzept für das Lernen im Kontext der zunehmenden Digitalisierung [KM16] vor und erklärt dieses zu einem integralen Bestandteil des Bildungsauftrages bereits in den Schulen der Primarstufe. Für die unmittelbare Lebensumwelt der Kinder in der Grundschule kommt insbesondere dem Mobiltelefon, bzw. in zunehmendem Maße dem Smartphone, eine wichtige Bedeutung zu. Zum Verstehen und Reflektieren der Funktionsweise von Mobiltelefonen gehören mittlerweile auch informatische Grundkompetenzen wie das Kennen und Verstehen von grundlegenden Prinzipien der digitalen Welt. Im vorliegenden Beitrag wird ein Konzept für ein Unterrichtskonzept zum Thema „Handy in der Grundschule“ vorgestellt.

2 Das Konzept der Handykiste

Bei der Handykiste in der Grundschule handelt es sich um eine Unterrichtskiste, die vielfältige Materialien rund um das Thema Handy und Mobilfunk enthält. Sie ist gedacht für einen handlungsorientierten Sachunterricht in der Grundschule. Die Lernziele dieses Unterrichts bestehen darin, die Schüler/innen der dritten und vierten Klasse dabei zu unterstützen, ein Verständnis für Mobiltelefone als Informatiksysteme und die Bedeutung dieser Informatik im Alltag zu entwickeln. Die Kiste enthält didaktisch aufbereitete Materialien für den Einsatz im Unterricht, die je nach Alter und Leistungsniveau der Schüler/innen ausgewählt und eingesetzt werden können, wie z.B. ein Mobiltelefon zum Auseinanderschrauben, Bastelanleitungen zum Basteln eines Dosentelefon und

¹ Carl von Ossietzky Universität, Department für Informatik, Abt. Didaktik der Informatik, Uhlhornsweg 84, 26129 Oldenburg, wibke.duwe@uni-oldenburg.de

Modellfiguren zum Erklären und Erfahrbarmachen eines Handyfunknetzes.

Der Inhalt der Handykiste ist in folgende Lehreinheiten gegliedert:

1. **Die Geschichte des Mobiltelefons:** Im ersten Kapitel der Handykiste wird die technische Entwicklung von Mobiltelefonen nachgezeichnet. Die Kinder sollen eigenständig Vorgängermodelle wie z.B. ein Dosentelefon bauen und erfahren, wie es zu der Erfindung eines kabellosen Telefons kam.
2. **Bestandteile eines Mobiltelefons:** In der zweiten Lehreinheit sollen Schüler/innen lernen, wie ein Mobiltelefon aufgebaut ist, welche Bestandteile es besitzt und diese auch benennen können. Praktisch erfahrbar wird dies u.a. durch das Aufschrauben eines Handys.
3. **Funktionsweise eines Mobiltelefons:** Im dritten Kapitel geht es um die Funktionsweise von Mobiltelefonen. Dabei stehen das Anrufen und das Versenden von Kurznachrichten über den Mobilfunk im Fokus. Mithilfe von geeigneten Materialien wie wasserdichten Behältern, Alufolie, Tüten etc. können die Schüler/innen eigene Experimente zur Funktionsweise von Mobilfunk durchführen.
4. **Weitere Dienste beim Smartphone:** Über den reinen Mobilfunk hinaus bieten Smartphones heute viele weitere Funktionen. Als besonderes wichtig ist hier die Internetverbindung hervorzuheben. Weitere insbesondere für Kinder wichtige und vielgenutzte Funktionen eines Smartphone stellen u.a. Handyspiele, Apps, Instant Messenger, Musikplayer, Kamera oder Ortungsdienste/GPS dar [MP14].
5. **Handysicherheit:** Im fünften Kapitel der Handykiste sollen Schüler/innen unterschiedliche Möglichkeiten der Verschlüsselung von Nachrichten kennenlernen. Risiken und Gefahren, die sich aus der Sammlung mobiler Daten (Stichwort Big Data) ergeben, sollen ebenfalls thematisiert werden.
6. **Die Zukunft von Mobiltelefonen:** Im abschließenden Kapitel soll die Zukunft der Handys daher thematisiert werden. Dieses Thema ermöglicht einen besonders kreativen Zugang, die Kinder können beispielsweise selbst aktiv werden und Handys der Zukunft und mögliche Funktionen erfinden und diese selbst basteln und präsentieren.

3 Literatur

- [KM16] KMK - Kultusministerkonferenz (Hrsg.) 2016: Bildung in der digitalen Welt. Strategie der Kultusministerkonferenz, <https://www.kmk.org/aktuelles/thema-2016-bildung-in-der-digitalen-welt.html>, 14.02.2017.
- [MP14] mpfs - Medienpädagogischer Forschungsverbund Südwest 2014: KIM-Studie 2014. Kinder + Medien, Computer + Internet. Basisuntersuchung zum Medienumgang 6 – 13-Jähriger, https://www.mpfs.de/fileadmin/files/Studien/KIM/2014/KIM_Studie_2014.pdf, 14.02.2017.

„Informatik entdecken – mit und ohne Computer“

**Mit ihrem Bildungsangebot unterstützt die Stiftung „Haus der kleinen Forscher“
informatische Bildung von drei- bis zehnjährigen Kindern**

Christine Günther¹

Abstract: Die Stiftung „Haus der kleinen Forscher“ stellt ein Bildungsangebot zur Verfügung, das pädagogische Fach- und Lehrkräfte in Kita, Hort und Grundschule in ihrer pädagogischen Arbeit im MINT-Bereich unterstützt. Seit 2016 widmet sich die Stiftung der fachlichen Fundierung des Inhaltsbereichs Informatik und entwickelte darauf aufbauend ein eigenes Bildungsangebot. Die Basis für die inhaltliche Entwicklung des Themas bildeten Zieldimensionen informatischer Bildung im Elementar- und Primarbereich.² Im Beitrag werden die Ziele der Stiftung, die den Themenbereich Informatik strukturierenden Inhalts- und Prozessbereiche, das Fortbildungskonzept, die pädagogischen Materialien sowie die Pilotierung des Angebots der Stiftung vorgestellt.

Keywords: Informatische Bildung, frühe informatische Bildung, Fortbildungskonzept, pädagogische Materialien

1 Ziele der Stiftung „Haus der kleinen Forscher“

Das Ziel der Stiftung „Haus der kleinen Forscher“ im Bildungsbereich Informatik ist es, Kindern zwischen drei und zehn Jahren erste Erfahrungen zum Thema Informatik zu ermöglichen. Mit ihren Angeboten möchte die Stiftung pädagogische Fach- und Lehrkräfte dabei unterstützen, Mädchen und Jungen in ihrer Auseinandersetzung mit informatischen Fragen, Problemen und Zielen zu begleiten.

2 Angebotsentwicklung

Ausgehend von Zieldimensionen informatischer Bildung für den Elementar- und Primarbereich³ wurden folgende Inhaltsbereiche im Angebot „Informatik entdecken – mit und ohne Computer“, konkret im Fortbildungskonzept und den pädagogischen Materialien, integriert: Die Bedeutung von Informatik im Alltag, Informationen und Daten, Planung, Steuerung und Optimierung.

¹ Stiftung „Haus der kleinen Forscher“, Abteilung: Inhaltliche Entwicklung, Rungestraße 18, 10179 Berlin, christine.guenther@haus-der-kleinen-forscher.de

² Bergner, N. et al., (in Vorbereitung)

³ ebd.

Prozessbezogene Aspekte informatischer Bildung wurden als didaktisch-methodisches Material als „Informatikkreis“⁴ für die Pädagoginnen und Pädagogen entwickelt. Dort wird informatisches Denken und Handeln als exploratives und gestaltendes, zyklisches Vorgehen dargestellt.⁵

Die Fortbildung ist nach den Empfehlungen aus den Zieldimensionen informatischer Bildung für den Elementar- und Primarbereich⁶ so konzipiert, dass die Pädagoginnen und Pädagogen das Thema sowohl ganz ohne Computer („unplugged“), mit dem Computer (softwarebasiert) und am Beispiel von kindgerechten Robotersystemen (Robotik) selbst entdecken und erforschen können. Für die Zielgruppe entscheidend ist die Erkenntnis, dass Grundprinzipien und Vorgehensweisen der Informatik auch mit Papier und Stift, mit Alltagsmaterialien oder mit reinem Körpereinsatz für Kinder erfahrbar werden können.⁷ Die Pädagoginnen und Pädagogen erhalten in der Fortbildung vielfältige Anregungen zur Umsetzung des Themas mit den Kindern, die sie selbst sofort erproben können. Außerdem erhalten die Teilnehmenden das pädagogische Materialpaket. In diesem finden sie viele der Praxisideen aus der Fortbildung auf Entdeckungskarten für Pädagoginnen und Pädagogen, Entdeckungskarten für Kinder und in einer Themenbroschüre festgehalten. Letztere bietet zudem fachliches als auch didaktisch-methodisches Hintergrundwissen. Das Bildungsangebot wird durch zwei digitale Lernspiele für Kinder im Grundschulalter (Ronjas Roboter und Fabios Flächen) sowie einen Onlinekurs für Pädagoginnen und Pädagogen ergänzt. In der Pilotphase (März bis August 2017) wurden sowohl die Fortbildung als auch die Materialien durch pädagogische Fach- und Lehrkräfte getestet und Überarbeitungen vorgenommen.

Literaturverzeichnis

- [BeiV] Bergner, N., Köster, H., Magenheimer, J., Müller, K., Romeike, R., Schulte, C. & Schroeder, U.: Zieldimensionen informatischer Bildung im Elementar- und Primarbereich. In Stiftung Haus der kleinen Forscher (Hrsg.), Frühe informatische Bildung – Ziele und Gelingensbedingungen für den Elementar- und Primarbereich. Wissenschaftliche Untersuchungen zur Arbeit der Stiftung „Haus der kleinen Forscher“ (Band 9). Verlag Barbara Budrich, Opladen, Berlin, Toronto, (in Vorbereitung).
- [In17] Stiftung „Haus der kleinen Forscher“ Informatikkreiskarte, 2017.
- [Sc17] Schulte, C., Magenheimer, J., Müller, K., Budde, L.: The Design and Exploration Cycle as Research and Development Framework in Computing Education. In: Beiträge zur EDUCON; Global Engineering Education Conference, IEEE, 2017.
- [BWF15] Bell, T., Witten, I., Fellows, M., 2015. cs unplugged, verfügbar unter <http://csunplugged.org/books/>

⁴ Stiftung „Haus der kleinen Forscher“, 2017

⁵ Schulte, C. et al., 2017

⁶ Bergner, N., et al., (in Vorbereitung)

⁷ Bell, T. et al., 2015

Informatische Grundbildung: Exploration des Erstzugangs zu Lego® Wedo 2.0 und Cubelets von Drittklässler_innen

Lennart Goecke,¹ Jurik Stiller,² Detlef Pech,³ Niels Pinkwart⁴

Abstract: Das vorgestellte Projekt geht der übergeordneten Frage nach, wie Kinder Algorithmen verstehen. Neben der didaktischen Begründung informatischer Bildung im Sachunterricht, werden erste Ergebnisse einer ersten Erhebung präsentiert. Die per Video erhobenen Daten basieren auf Explorationen von Kindern, die erstmals mit Lego® WeDo 2.0 und Cubelets arbeiten.

Keywords: Sachunterricht; Grundschule; Algorithmusverständnis; Lebensweltzugang

1 Einleitung & Stand der Forschung

Erkenntnisleitend für das hier skizzierte Projekt ist die übergeordnete Frage nach dem Verhältnis von Informatik, informatischer Bildung und Sachunterricht. Im Fokus stehen dabei Fragen, die die theoretisch und empirisch begründete Entwicklung und Evaluation von Unterricht betreffen. Materialien wie Lego® WeDo 2.0 und Cubelets adressieren tendenziell alle Grundschüler_innen und nicht nur Interessierte. Daher eignen sie sich als ein Zugang zum Konstrukt *Algorithmusverständnis von Grundschüler_innen*. Fachdidaktisch lässt sich die Relevanz von Informatik-Themen als Teil von Sachunterricht gut begründen. In dessen Didaktik werden Klafkis allgemeindidaktische Aspekte zur Bedeutung von Inhalten weit rezipiert, sodass sich bspw. *Grundbildung zum Algorithmusbegriff* u. a. gut als zukunftsbedeutsam begründen lässt.

Welche Effekte beim Umgang mit algorithmischen Phänomenen in der Grundschule auftreten und wie sich diese Lernprozesse beschreiben lassen, wird bislang nur wenig und nur anhand von Teilaspekten untersucht [Is15, S. 131f. SRV16; KG14; Se11]. Dies ist jedoch eine Voraussetzung zur Entwicklung und Evaluation von Unterricht. Die Verortung des Algorithmusverständnis in der informatischen (Grund-) Bildung begründet sich durch deren alltägliche Relevanz.

Daher lautet die leitende Fragestellung dieser Untersuchung: *Wie verstehen Grundschüler_innen Algorithmen?* Der Beantwortung wird sich in einem ersten Schritt mittels einer

¹ Humboldt-Universität zu Berlin, Sachunterrichtsdidaktik und seine Didaktik, Friedrichstr. 194, 10117 Berlin, lennart.goecke@hu-berlin.de

² gleiche Affiliation, jurik.stiller@hu-berlin.de

³ gleiche Affiliation, detlef.pech@hu-berlin.de

⁴ Humboldt-Universität zu Berlin, Inst. f. Informatik, Rudower Chaussee 25, 12489 Berlin, pinkwart@hu-berlin.de

explorativen Analyse angenähert. Dazu wurden Unterrichtssituationen hergestellt, in denen der hohe Aufforderungscharakter der Materialien genutzt wird um Explorieren anzuregen. Das Material ermöglicht dabei grundsätzlich, intuitiv Algorithmen zu entwickeln und zu testen. So wurden mehrere Drittklässler_innen bei dem ersten Umgang mit Lego® WeDo 2.0 und Cubelets gefilmt. Aus der Analyse der Daten lassen sich verschiedene Erkenntnisse zu erreichbaren Kompetenzen und notwendigen didaktischen Entscheidungen ableiten. Diese können für weitere Untersuchungen sowie didaktische Situationen grundlegend sein. Erste Ergebnisse weisen u. a. auf verschiedene Typen des Zugangs zu den Materialien hin. Ein zu beobachtendes Handlungsmuster ist die (technisch im Ergebnis irrelevante) wiederholte Verwendung eines Programmblockes, um ein Lego® Gefährt schneller werden zu lassen. Der unangeleitete Umgang mit Cubelets hingegen lässt daneben Rückschlüsse auf kindliche Verständnisweisen von ihnen unbekanntem technischen Systemen zu. Aus den Daten lässt sich dabei schließen, dass es Kindern gelingen kann, durch gezielte Manipulationen belastbare Hypothesen über Funktionen der Cubelets aufzustellen.

2 Ausblick

Bei dem Projekt handelt es sich um eine Voruntersuchung im Rahmen eines Dissertationsvorhabens. Im weiteren Verlauf werden präzisere Rückschlüsse auf genuin kindliche Zugänge zu algorithmischen Phänomenen erarbeitet. Hierfür lässt sich auf die ersten Ergebnisse aufbauen, indem Unterrichtssettings gezielt typische Vorstellungen berücksichtigen, um Konzeptwechsel zu ermitteln. So lassen sich etwa erreichbare Kompetenzen beschreiben und auswerten. Perspektivisch könnte dazu auch der Frage *Wie hängen Algorithmusverständnis und scientific literacy zusammen?* nachgegangen werden, da es sich hierbei um ein in der Sachunterrichtsdidaktik zentrales Konzept handelt und hoher Bedarf besteht, einzelne Facetten konkreter zu operationalisieren.

Literatur

- [Is15] Israel, M.; Pearson, J. N.; Tapia, T.; Wherfel, Q. M.; Reese, G.: Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education* 82/, S. 263–279, 2015.
- [KG14] Kalelioglu, F.; Gülbahar, Y.: The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education* 13/, 2014.
- [Se11] Serafini, G.: Teaching Programming at Primary Schools: Visions, Experiences, and Long-Term Research Prospects. In (Kalas, I.; Mittermeir, R. T., Hrsg.). Springer, Berlin, Heidelberg, S. 143–155, 2011.
- [SRV16] Sáez-López, J.; Román-González, M.; Vázquez-Cano, E.: Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education* 97/, 2016.

Kindliche Modelle der digitalen Welt

Anja Gärtig-Daug¹, Katharina Weitz¹, Ute Schmid¹

Abstract: Digitale Medien prägen heutzutage unser alltägliches Leben. Diesem Einfluss sind auch Kinder ausgesetzt, selbst wenn sie selbst noch keine Medien aktiv nutzen. Die Vermutung, dass Kinder sich gedanklich mit dem Thema beschäftigen, liegt daher nahe. Der vorliegende Beitrag untersucht, über welche mentalen Modelle zur Funktionsweise digitaler Medien Kinder im Grundschulalter verfügen.

Keywords: mentale Modelle, digitale Medien, Grundschule

1 Einleitung und Motivation

Technik begegnet Kindern in einer hochkomplexen Form, die einen aktiven und verstehenden Zugang erschwert [NH13]. Kinder konzipieren jedoch vereinfachte mentale Modelle, die ihnen als Grundlage zum Verstehen von physikalischen oder technischen Prozessen dienen [Ge02]. Die vorliegende Studie untersucht mentale Modelle von Grundschulkindern zur *Digitalen Repräsentation* und zu *Algorithmen*.

2 Empirische Studie und Ergebnisse

Die Untersuchung wurde im Rahmen eines Informatikprojekts² mit den Klassen 2 bis 4 (Alter der Kinder zwischen 7 und 10 Jahren) einer Grundschule durchgeführt. Aus der 2. Klasse wurden 15 Kinder (10 Mädchen, 5 Jungen), aus der 3. Klasse 17 Kinder (8 Mädchen, 9 Jungen) und aus der 4. Klasse 15 Kinder (6 Mädchen, 9 Jungen) einbezogen.

Im Zentrum der empirischen Untersuchung standen die Fragen, welche informatischen Fachbegriffe Grundschulkindern bekannt sind und ob Kinder in diesem Alter über mentale Modelle zu den Funktionsprinzipien von digitalen Medien verfügen.

Von den Schülerinnen und Schülern wurde erfragt, ob sie die Begriffe „Pixel“, „digital“ und „Algorithmus“ kennen. Zusätzlich wurde erhoben, ob die Kinder einen Unterschied zwischen dem Sortieren bei Menschen und bei Computern kennen. Tab. 1 gibt einen Überblick über die Anzahl der Kinder, die glaubten, den Begriff bzw. die Antwort zu kennen, sowie eine Auswahl der genannten Beschreibungen.

¹ Otto-Friedrich-Universität Bamberg, Fakultät Wirtschaftsinformatik und Angewandte Informatik, Forschungsgruppe Elementarinformatik, An der Weberei 5, 96045 Bamberg, anja.gaertig-daug@uni-bamberg.de, katharina-blandina.weitz@stud.uni-bamberg.de, ute.schmid@uni-bamberg.de

² <http://nachwuchs.wiai.uni-bamberg.de/i4kids.html>

Klasse	Beiträge	Exemplarische Beschreibungen
Pixel		
2	6	Computerbilder bestehen aus 1.000/vielen Pixeln (3x)
3	5	beim Computerspielen, da ist es häufig pixelig
		wenn man Fotos auf dem Computer speichert; ist sehr klein, nicht erkennbar; ein großes Pixel enthält 100 Minipixel
4	4	minimal kleines Viereck (2x)
		Pixel sind beim Computer / Fernseher; bei schlechtem Empfang
Digital		
2	3	digital ist, wenn man Fernsehen/Handy guckt (2x)
3	2	z. B. muss man den Ton am Laptop digitalisieren
4	2	weit entfernt / nicht live gesendet
Algorithmus		
2	3	Krankheit (2x)
3	2	ein Takt / hat etwas mit einer Uhr zu tun
4	1	Krankheit
Unterschied zwischen dem Sortieren bei Menschen und bei Computern		
2	7	Computer sortiert schneller /sortiert mit Kamera
3	5	Computer wissen sehr viel und vergessen es nicht (2x)
		Computer kann mehr Dinge vergleichen; kann Dinge ersetzen
4	5	Computer hat Speicher (2x); Zahlen/ABC/Spiele sind gespeichert
		Computer sortiert blitzschnell, sortiert nach Daten, ABC
		beim Computer ist das Sortieren leichter, weil er nicht so groß ist

Tab. 1: Ergebnisse der Grundschulerhebung zum informatischen Vorwissen von Kindern

3 Diskussion und Ausblick

Die Ergebnisse legen nahe, dass Kinder bereits in einem sehr jungen Alter mentale Modelle zur digitalen Welt aufbauen. Die Ausdifferenzierung und Korrektheit scheint nicht unbedingt vom kognitiven Entwicklungsstand, sondern vom Interesse der Kinder am Thema abzuhängen. Es bleibt zu untersuchen, inwieweit informatische Bildungsangebote in Vor- und Grundschule zu einer korrekten Ausdifferenzierung mentaler Modelle beitragen können.

Literaturverzeichnis

- [Ge02] Gentner, D.: Mental models, Psychology of. In Smelser, N; Bates, P. (Hrsg.): International Encyclopedia of the Social and Behavioral Sciences, Elsevier Science, Amsterdam, S. 9683-9687), 2002.
- [NH13] Niggler, A.; Holl, P.: Perspektiven naturwissenschaftlicher Bildung in Kindergarten und Grundschule. Plus Lucis, 1-2, S. 31-33, 2013.

Lernwelten verbinden - Coding und Making mit Minecraft

Mirek Hančl¹

Abstract: Minecraft ist ein populäres, mehrspielerfähiges Videospiel. Im Bildungskontext wird Minecraft verstärkt eingesetzt, um Fachinhalte und prozessbezogene Kompetenzen gleichermaßen und zielgruppengerecht zu vermitteln. Die Automatisierung virtueller Schildkröten und das Ansteuern virtueller Logikschaltungen in der mehrspielerfähigen Lernumgebung vermitteln den Schülern grundlegende Kenntnisse in der Programmierung. Die Ansteuerung realer Hardware aus Minecraft heraus verbindet die virtuelle Lernwelt der Schüler mit ihrer realen Lebenswelt und fördert in kreativer Art und Weise das Machen. Die Schüler lernen dabei im Sinne einer interaktionistisch-konstruktivistischen Didaktik und des connected learning approach.

Keywords: Connected Learning, Konstruktivismus, Minecraft, Robotik, Algorithmik, Coding, Making

1 Das Videospiel Minecraft

Minecraft ist ein populäres, mehrspielerfähiges Videospiel, in dem in einer dreidimensionalen Welt aus Blöcken gebaut werden kann. Es gibt kein primäres Spielziel, da Minecraft dem Genre der open world games zuzuordnen ist.

2 Minecraft im Bildungskontext

Minecraft wird seit einigen Jahren verstärkt im Bildungsbereich eingesetzt. Nahezu jede Verwendung von Minecraft im Unterricht macht sich den Vorteil des Mehrspielermodus zunutze, wodurch die gesamte Lerngruppe gemeinsam in der virtuellen Welt interagiert. Dadurch werden die didaktisch-methodischen Möglichkeiten aus einer realen Lernumgebung wie einem Unterrichtsraum verlagert in die virtuelle Lernumgebung [Ha16]. Unterricht mit Minecraft wird idealerweise durch eine interaktionistisch-konstruktivistische Didaktik inspiriert. Diese sieht Lernprozesse nicht nur durch die individuelle Wahrnehmung der Welt durch den Lernenden beeinflusst, sondern auch durch sein soziokulturelles Umfeld, in dem der Lernprozess stattfindet [Re12]. Insbesondere durch den Einsatz digitaler, vernetzter Medien in der Projektarbeit folgt

¹ Universität Osnabrück, Institut für Informatik, AG Didaktik der Informatik, Wachsbleiche 27, 49090 Osnabrück, mirek.hancl@uni-osnabrueck.de

Coding und Making mit Minecraft einem besonderen pädagogischen Ansatz, dem connected learning [Co15]. Denn digitale Medien ermöglichen erst das einfache, vielfältige Produzieren, Teilen und Konsumieren multimodaler Produkte, was sich im Sinne des connected learning besonders motivierend auf Lerner auswirkt [It13].

3 Minecraft im Kontext der informatischen Bildung

3.1 Coding mit Minecraft

Schon mit dem klassischen Spiel Minecraft lassen sich die Themen Boolesche Algebra, Wahrheitswerte und Logikgatter anschaulich vermitteln. In Minecraft ist auch ohne Erweiterung ein Set an Eingabe- und Ausgabeelementen verfügbar, mit denen einfache bis komplexe virtuelle Schaltkreise konstruiert werden können. Durch die Erweiterung des Spiels mit ComputerCraftEdu kann Schülern Algorithmik in motivierender Weise vermittelt werden. ComputerCraftEdu erweitert Minecraft um virtuelle Schildkröten, die sowohl mit einer visuellen als auch textuellen Programmiersprache programmiert werden können (neudeutsch: Coding).

3.2 Making mit Minecraft

Zeitgemäßer Informatikunterricht tut gut daran, wenn er das aktive Tun sowohl im realen (analogen) Raum als auch in der virtuellen Welt der digitalen Medien fordert und fördert. Er fördert kritisches Lernen dann am besten, wenn er beide Welten im Sinne eines immersiven Lernprozesses, einer erweiterten Lernwelt, miteinander verknüpft und dem Lerner die Anknüpfungspunkte bewusst aufzeigt. In Minecraft können Projekte zu den Themenbereichen Informatiksysteme sowie Informatik, Mensch und Gesellschaft mit den Erweiterungen RealRobots und bitCraft anschaulich umgesetzt werden. Die Ansteuerung realer Hardware wie Arduino, Lego WeDo, RaspberryPi und littleBits aus Minecraft heraus verbindet die virtuelle Lernwelt der Schüler mit ihrer realen Lebenswelt und fördert in kreativer Art und Weise das Machen (Neudeutsch: Making).

Literaturverzeichnis

- [Co15] Connected learning through Minecraft, <http://www.informatics.uci.edu/connected-learning-through-minecraft/>, geprüft am 12.02.2017.
- [Ha16] Hanel, M.: Minecraft als Lernumgebung. Computer+Unterricht 102, S. 28-30, 2016.
- [It13] Ito, M. et al: Connected Learning. An Agenda for Research and Design. Digital Media and Learning Research Hub, Irvine, CA, 2013.
- [Re12] Reich, K.: Konstruktivistische Didaktik. Das Lehr- und Studienbuch mit Online-Methodenpool. Beltz Pädagogik, Weinheim, Bergstr., 2012.

Motivierung im Informatikunterricht: eine quantitative Online-Studie zu Planungsgrundlagen von Lehrpersonen

Stefanie Jäckel¹

Abstract: Das didaktische Prinzip Motivierung erweist sich als besonders bedeutsam, weil motivierte und interessierte Schülerinnen und Schüler leichter lernen [Br12]. Da Erkenntnisse über Auswahl Faktoren der Lehrpersonen für eine bestimmte Motivierung und die zur Entwicklung von Motivierungen benutzten Informationsquellen und Materialien ein Desiderat informatikdidaktischer Forschung sind, widmet sich die vorliegende Studie diesem Untersuchungsgegenstand.

Keywords: Motivierung, Informatikunterricht, Unterrichtseinstieg, Lehrerhandeln

1 Design der Erhebung und Beschreibung der Stichprobe

Mit Hilfe von Mehrfachauswahl-Fragetypen mit Ergänzungsmöglichkeit wurden sowohl Entscheidungsgrundlagen der Lehrpersonen als auch verwendete Quellen und Materialien erfasst.² Die in den Fragen vorgegebenen Items wurden aus einer explorativen Studie [Jä15], theoretischen Modellen [Ke10] und von der Autorin zum Thema durchgeführten Workshops abgeleitet. An der Studie beteiligten sich von 06 bis 09/2016 156 Informatiklehrkräfte aus Deutschland, Österreich und der Schweiz. Die Fachlehrerinnen (37,4 %) und -lehrer (62,6 %) unterrichten schwerpunktmäßig in der Sekundarstufe II (54,2 %). Die häufigsten (kombinierten) Studienfächer sind Informatik (m: 71,4%, w: 69,2%); Mathematik (m: 65,3%, w: 67,3%) und Physik (m: 33,7%, w: 23,1%). Bezüglich der Unterrichtserfahrung (inklusive Referendariat) ist die Zusammensetzung der Stichprobe sehr gemischt, woraus eine gewisse Repräsentativität hinsichtlich der Zusammensetzung der Grundgesamtheit geschlussfolgert werden kann. Die Art der Durchführung der Untersuchung setzt bei den Lehrkräften Professions- und Reflexionswissen über ihr Lehrerhandeln voraus.

2 Für Motivierungen benutzte Quellen und Materialien

Die teilnehmende Lehrerschaft benutzt für die Konzeption ihrer motivierenden Unterrichtseinstiege am häufigsten das Internet (n=123) als Informations- und Materialquelle, gefolgt von selbst entwickelten Materialien (n=106) sowie Ideen/Konzepten aus Fortbil-

¹ Friedrich-Schiller-Universität Jena, Fakultät für Mathematik und Informatik, Ernst-Abbe-Platz 2, 07743 Jena, stefanie.jaekel@uni-jena.de

² Detaillierte Untersuchungsergebnisse unter: <http://www.informatikideen.de/motivierung.html>

dungsveranstaltungen (n=86). Gruppieren man die Items, kann gezeigt werden, dass Lehrkräfte mit 47,6 % im Mittel am häufigsten durch Erfahrungswerte (eigene Materialien, Ideen/Konzepte aus Fortbildungen, Tipps von Kollegen, an der Schule verfügbare Materialien) zu ihren Motivierungen gelangen. Außerdem sind Medien allgemein (Internet, Software, Tageszeitungen/Zeitschriften) mit 35,5% für die Ideenfindung der Lehrkräfte noch ausschlaggebender als (fach)wissenschaftliche Forschung (Informatikfachbücher und -zeitschriften, didaktische Literatur, pädagogische Literatur). Je nach Unterrichtserfahrung der Teilnehmer verändern sich auch Planungsgewohnheiten: so benutzen Lehrkräfte mit über 15 Jahren Lehrtätigkeit auch beim Motivieren an erster Stelle ihre eigenen Materialien, was vermuten lässt, dass sich Lehrkräfte ein Repertoire an Motivierungen zulegen, das sie in der Praxis testen, erweitern und regelmäßig nutzen.

3 Entscheidungsgrundlagen für motivierende Einstiege

Die meisten Lehrpersonen geben an, dass sie bei Entscheidungen für oder gegen einen motivierenden Einstieg vor allem Problemstellungen aus der Lebenswelt (57,4%) sowie Interessen der Schüler einbeziehen (50,3%) und sich an Informatikanwendungen im Alltag (51,0%) sowie dem Vorwissen der Schüler (44,5%) orientieren.

Die männlichen Teilnehmer berücksichtigen bei der Entscheidung für eine Motivierung stärker als die Frauen eigene fachliche Interessengebiete (m: 36,7%, w: 20,7%) und aktuelle Ereignisse (m: 35,7%, w: 25,9%). Lehrerinnen orientieren sich stärker an räumlichen Gegebenheiten (m: 10,2%, w: 24,1%), der Vernetzung zu anderen Unterrichtsinhalten (m: 26,5%, w: 34,5%) und Informatikanwendungen im Alltag (m: 48,0%, w: 55,2%).

Um zu untersuchen, welche Grundlagen die Entscheidung für eine Motivierung am stärksten beeinflussen, wurden die Items den Faktoren *Schüler*, *Fach Informatik* und *Organisatorischem* zugeordnet. Bei der Entscheidung für einen motivierenden Einstieg richten sich Informatiklehrkräfte am meisten nach ihren Schülern (45,3%). Danach berücksichtigen sie fachliche Eigenheiten (30,7%) und organisatorische Gegebenheiten (23,1%).

4 Literaturverzeichnis

- [Br12] Brohm, M.: Motivation lernen. Das Trainingsprogramm für die Schule. Beltz, Weinheim, 2012.
- [Jä15] Jäckel, S.: Schüler für Fachthemen interessieren und motivieren - Informatikunterricht im Fokus Informatik und Schule, In (Jens Gallenbacher Hrsg.): Informatik allgemeinbildend begreifen, INFOS 2015, Darmstadt, Germany. GI, 2015; S. 171–180.
- [Ke10] Keller, J. M.: Motivational design for learning and performance. The ARCS model approach. Springer, New York, London, 2010.

Entwicklung eines Online-Tools zur Bestimmung objektorientierter Programmierkompetenzen

Matthias Kramer¹, Vahid Samimi² und Torsten Brinda³

Abstract: Basierend auf einem theoretisch hergeleiteten Kompetenzmodell für objektorientierte Programmierung wird ein Prototyp eines Messwerkzeuges zur empirischen Bestimmung solcher Kompetenzen vorgestellt. Erste Tests existieren bereits, zukünftige Tests werden skizziert.

Keywords: Objektorientierung, Kompetenzmessung, Messinstrument.

1 Einleitung und Hintergrund

Bedingt durch die Ergebnisse internationaler Vergleichsstudien [MN13] sind die normative Bestimmung und empirische Überprüfung von Kompetenzen Themen, denen sich die Forschung im Bereich der Erziehungswissenschaft und Psychologie in den letzten Jahren ausgiebig gewidmet hat [BGS15]. Im Bereich der objektorientierten Programmierung fokussiert sich das Projekt *COMMOOP* auf die Identifikation ebensolcher Kompetenzen. In dessen Rahmen entsteht zurzeit ein Instrument, welches sich sowohl für Vergleichsstudien großer Kohorten als auch für individualdiagnostische Zwecke eignen wird.

Basierend auf gängigen Definitionen des Begriffs *Kompetenz*, siehe [K108], werden Kompetenzen im Bereich der objektorientierten Programmierung in diesem Zusammenhang als domänenspezifische kognitive und metakognitive Fähigkeiten und Fertigkeiten verstanden, die Individuen in die Lage versetzen, Probleme im Bereich der objektorientierten Programmierung zu lösen. Im Projekt *COMMOOP* ist im Rahmen einer intensiven Literaturrecherche dabei ein vierdimensionales Kompetenzmodell für objektorientierte Programmierung entstanden [KHB16], Testitems wurden auf dessen Basis beispielhaft abgeleitet [KTB16a] und ein erster Subskalentest mit zwölf Items erstellt [KTB16b]. Die Durchführung mittels Stift und Papier hat sich dabei jedoch bzgl. Auswertbarkeit und Erreichbarkeit von Testpersonen als nachteilig erwiesen. Daher wurde damit begonnen, eine elektronische Version des Testinstrumentes zu erstellen, die im weiteren Verlauf des Projektes sowohl zum summativen als auch zum formativen Assessment genutzt werden soll.

¹ Universität Duisburg-Essen, Didaktik der Informatik, Schützenbahn 70, 45127 Essen, matthias.kramer@uni-due.de

² Universität Duisburg-Essen, Didaktik der Informatik, Schützenbahn 70, 45127 Essen, vahid.samimi@stud.uni-due.de

³ Universität Duisburg-Essen, Didaktik der Informatik, Schützenbahn 70, 45127 Essen, torsten.brinda@uni-due.de

2 Implementierung, bisheriger Stand und Ausblick

Um eine Installation des Messwerkzeugs auf Schulrechnern zu vermeiden, fiel die Entscheidung auf eine skriptbasierte Webimplementierung. Nach Besuchen eines entsprechenden Links werden die Probanden schrittweise durch die einzelnen Tests geführt. Zu Beginn geben sie entsprechende soziodemographische Daten ein (Alter, Geschlecht, Vorerfahrungen etc.), welche am Ende mit der erhobenen Leistung korreliert werden. Diese werden über eine HTTPS-Verbindung weitergeleitet und in einer Datenbank auf Servern der Universität Duisburg-Essen abgelegt. Anschließend werden die Testpersonen gebeten, die angezeigten Aufgaben nacheinander zu bearbeiten. Zeitgleich zur Bearbeitung werden den bisherigen Ergebnissen Skalenwerte der vorher postulierten Kompetenzdimensionen zugeordnet. Am Ende erhalten die Testpersonen eine individuelle Übersicht über Stärken und Schwächen in den einzelnen Kompetenzdimensionen.

Basierend auf dem vorher genannten Kompetenzmodell werden momentan schrittweise Tests entwickelt, die sich jeweils einzelnen Kompetenzfacetten zuordnen lassen. Es existieren Implementierungen für das Markieren objektorientierter Konzepte in Quelltexten sowie für das Interpretieren von Quelltexten in Abhängigkeit von der Leistung des Arbeitsgedächtnisses. Diese wurden im Vorfeld bereits pilotiert. Im weiteren Verlauf werden diese schrittweise verbessert. Weitere Tests werden bspw. die Fähigkeiten des mentalen Ausführens von Quellcode (Tracing), des lückenweisen Auffüllens einzelner Schlüsselwörter sowie des vollständigen Ergänzens größerer Quelltextfragmente umfassen.

Literaturverzeichnis

- [BGS15] Blömeke, Sigrid; Gustafsson, Jan-Eric; Shavelson, Richard J: Beyond dichotomies. *Zeitschrift für Psychologie*, 2015.
- [KHB16] Kramer, Matthias; Hubwieser, Peter; Brinda, Torsten: A Competency Structure Model of Object-Oriented Programming. In: *Learning and Teaching in Computing and Engineering (LaTICE)*, 2016 International Conference on. IEEE, S. 1–8, 2016.
- [KI08] Klieme, Eckhard; Hartig, Johannes; Rauch, Dominique; Hartig, J; Klieme, E; Leutner, D: The concept of competence in educational contexts. *Assessment of competencies in educational contexts*, S. 3–22, 2008.
- [KTB16a] Kramer, Matthias; Tobinski, David A.; Brinda, Torsten: Modelling Competency in the Field of OOP: From Investigating Computer Science Curricula to Developing Test Items. In: *Stakeholders and Information Technology in Education*. International Federation for Information Processing, Springer, S. 37–46, 2016.
- [KTB16b] Kramer, Matthias; Tobinski, David A.; Brinda, Torsten: On the Way to a Test Instrument for Object-oriented Programming Competencies. In: *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*. Koli Calling '16, ACM, New York, NY, USA, S. 145–149, 2016.
- [MN13] Martens, Kerstin; Niemann, Dennis: When do numbers count? The differential impact of the PISA rating and ranking on education policy in Germany and the US. *German Politics*, 22(3):314–332, 2013.

Der Biber der Informatik in Österreich - Anmerkungen und Perspektiven

Peter Micheuz¹, Gerald Futschek²

Keywords: Informatik, Biber der Informatik, Wettbewerb, Statistiken, Aufgaben.

Der „Biber der Informatik“-Wettbewerb [DS16] fand 2016 in Österreich zum 10. Mal statt und wurde von der Österreichischen Computergesellschaft organisiert [OCG16]. Die bei diesem Online-Wettbewerb generierten administrativen und aufgabenbezogenen Daten lieferten aufschlussreiche Ergebnisse.

	Stufe	VS	AHS	NMS	BHS
Grundschule	3	1,2%			
	4	1,6%			
Sekundarstufe I Benjamin Meteor	5		9,8%	2,0%	
	6		9,9%	3,5%	
	7		12,6%	3,8%	
	8		10,0%	4,2%	
Sekundarstufe II Junior Senior	9		23,6%		3,5%
	10		4,5%		1,8%
	11		2,4%		1,8%
	12		1,6%		1,3%
	13				1,2%

Abb. 1: Schulstufen- und schultypenspezifische Übersicht

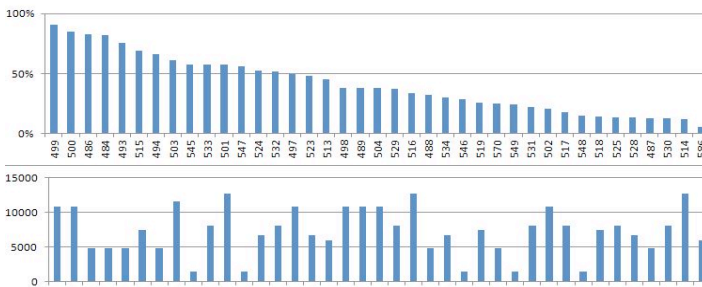


Abb. 2: Prozentuelle Lösungshäufigkeiten und korrespondierende absolute Teilnehmerzahlen

Beispielsweise zeigte sich auch 2016 wieder eine sehr heterogene quantitative Verteilung der „Biber der Informatik“-Teilnahme auf die österreichischen Schultypen Grundschule (VS), Sekundarstufe I und II (AHS, NMS, BHS), Schulen und Jahrgänge (Abb. 1). Mit diesem Wissen ist eine zielgerichtete Bewerbung in den kommenden Jahren möglich. Die ermittelten Lösungshäufigkeiten bei den Aufgaben legen den Schluss nahe, dass die Aufgaben im Schnitt schwierig waren (Abb. 2).

¹ Alpen-Adria-Universität Klagenfurt, Institut für Informatikdidaktik, Universitätsstraße 65-67, 9020 Klagenfurt, peter.micheuz@aau.at

² Technische Universität Wien, Institut für Softwaretechnik und Interaktive Systeme, Favoritenstraße 9/188, 1040 Wien, gerald.futschek@tuwien.ac.at

Das Biber-Jubiläumsjahr 2016 wurde auch durch empirische Umfragen begleitet. Eine davon war an die Biber-Organisatoren an den Schulen gerichtet und lieferte interessante Aufschlüsse im Hinblick auf Aufgabenstellungen, Organisation, Durchführung und Verbesserungswünsche. Dabei zeigte es sich unter anderem, dass

- über 70% bereits dreimal oder öfter den Informatikbiber organisiert haben,
- in ca. 1/3 der Schulen der Biber der Informatik auch in Klassen ohne formellen Informatikunterricht angeboten wurde,
- die niederländische Wettbewerbsplattform zu über 90% technisch gut funktioniert hat und es auch kaum technische Probleme (weniger als 10%) an der Schule gab,
- die Biber-Aufgaben 2016 zu über 90% als gut befunden wurden,
- ca. 30% die Aufgaben im Schnitt als zu schwierig empfanden und sich die Hälfte im Schnitt leichtere Aufgaben wünschen.

Die Entwicklungsperspektiven für den Biber der Informatik sind mannigfaltig. Der große Fundus an öffentlich zugänglichen Biber-Aufgaben, die Daten im Rahmen der Biber-Organisation an den Schulen und die Ergebnisse stellen gute Ausgangspunkte für viele Unterrichtsszenarien dar. Diese decken das breite Spektrum Digitaler Bildung von der Kerninformatik über Anwendungsorientierung bis hin zur Computer Literacy ab.

Der Aussage „Ich habe schulbezogene Ergebnisdaten nach dem Biber-Bewerb im Unterricht auswerten lassen“ wurde von einem Viertel der organisierenden Lehrkräfte zugestimmt. Abb. 3 zeigt zwei diesbezügliche konkrete Auswertungen von Schülern.

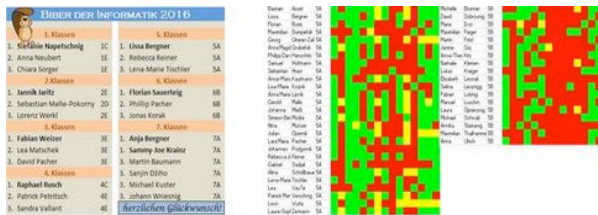


Abb. 3: Schülerarbeiten (Rangliste, Visualisierung zweier Klassenergebnisse)

Der Zugriff auf den Biber-Datenbestand (sowohl auf Schulebene als auch anonymisiert auf nationaler Ebene z.B. im Rahmen der Open Data Initiative) eröffnet ein weites praktisches und didaktisches Betätigungsfeld für viele Spielarten von Anwendersoftware.

Literaturverzeichnis

[DS16] Dagiene, V.; Sentance, S.: It’s Computational Thinking! Bebras Tasks in the Curriculum. In International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, Springer International Publishing, S. 28-39, 2016.

[OCG16] OCG, Österreichische Computergesellschaft, www.ocg.at/de/biber-der-informatik, Stand: 10.2.2017.

„Informatik im Unterricht – So geht’s“ – Informatik im Primarbereich

Martin Fricke, Kathrin Haselmeier, Ludger Humbert, Dorothee Müller, Daniel Siebrecht¹

Abstract: An Beispielen wird veranschaulicht, wie die Verzahnung von Prozess- und Inhaltsbereichen der Bildungsstandards Informatik für Schülerinnen und Schüler des Primarbereichs gestaltet wird.

Keywords: Informatik im Primarbereich, Grundschule, informatische Bildung, Bildungsstandards

1 Informatik im Primarbereich

Für zwei aktuelle Projekten zur informatischen Bildung im Primarbereich ist die Kompetenzebene eine zentrale Analyse- und Konstruktionsebene der angestrebten informatischen Bildung. In der Informatikdidaktik hat sich die Strukturierung der Bildungsstandards durch je fünf Prozess- und Inhaltsbereiche, die in den konkreten Kompetenzen verwoben sind, durchgesetzt. Dies wird in beiden Projekten für den Primarbereich adaptiert.

Arbeitskreis „Bildungsstandards Primarbereich“ (AK Primar) – GI

Der AK Primar des Fachausschusses Informatische Bildung in Schulen (FA IBS) der GI erarbeitet Bildungsstandards Informatik für den Primarbereich [GI17]. Damit werden die Anforderungen für informatische Allgemeinbildung für Schülerinnen und Schüler im Primarbereich auf Kompetenzebene ausgewiesen.

Projekt Informatik an Grundschulen (IaG) – Nordrhein-Westfalen

IaG [MS17] ist ein Pilotprojekt zur Erprobung von Konzepten zur Förderung der informatischen Bildung an Grundschulen. Es wird in Kooperation der jeweiligen Fachdidaktik Informatik der drei universitären Standorte RWTH Aachen, Universität Paderborn und Bergische Universität Wuppertal mit dem Schulministerium Nordrhein-Westfalen durchgeführt. An jedem Standort arbeiten zwei Grundschullehrkräfte. Gemeinsam mit den Fachdidaktiken erarbeiten sie themenspezifische Module (Unterrichtseinheiten), die den Schülerinnen und Schülern einen Einblick in die Informatik ermöglichen und sie bei der Entwicklung eines grundlegenden Verständnisses von zentralen Gegenständen der Informatik unterstützen.

¹ Grundschule Stübchen Solingen, Offene Ganztagschule Liegnitzerstr. Wuppertal, Bergische Universität Wuppertal, Didaktik der Informatik – {mfricke,khaselmeier,humbert,dmueller,siebrecht}@uni-wuppertal.de

2 Verzahnung von Prozess- und Inhaltsbereichen

Informatik im Primarbereich steht derzeit in den Anfängen der theoretischen und praktischen Umsetzung. Insbesondere ist Informatik derzeit kein verpflichtendes Element der Lehrerbildung. Damit fehlt sowohl die informatische Expertise in der Grundschule wie auch die Grundschulexpertise in der Informatik. Die Einbeziehung von Grundschullehrkräften in die Ausarbeitung der Bildungsstandards und die Vermittlung informatischer Grundlagen für Grundschullehrkräfte sind zwei Seiten der selben Medaille. Der AK Primar trägt dieser Verknüpfung Rechnung, dort wird gemeinsam fachwissenschaftlich und grundschuldidaktisch diskutiert. Parallel erschließen im Projekt IaG Grundschullehrkräfte grundlegende Aspekte der Informatik und entwickeln Ideen für die Umsetzung in der Grundschule. Aus diesen Arbeitsprozessen resultiert die Aufarbeitung der Prozess- und Inhaltsbereiche der Informatik in eine übersichtliche Darstellung für Lehrkräfte in der Primarstufe [Fr16].

Aus der Verbindung der Prozess- und Inhaltsbereiche entsteht die Kompetenzebene. Die Zuordnung erfolgt nicht in einer Eins-zu-eins-Zuordnung, damit die Fehlvorstellung vermieden wird, dass nur eine Kompetenz je Prozess- oder Inhaltsbereich konstruiert werden kann. Um zudem zu verdeutlichen, dass sich nicht in jeder der Kompetenzen genau ein Prozess- und ein Inhaltsbereich verbinden, wurden auch Kompetenzen gewählt, denen mehrere Bereiche zuzuordnen sind. Die Schülerinnen und Schüler erfahren die informatische Modellierung (inklusive der Implementierung), indem sie beispielsweise gemeinsam eine Steuerung für einen Roboter entwickeln. Der Inhaltsbezug ist dabei der Bereich Information und Daten.

Die Vielfalt der Möglichkeiten zur Verzahnung lässt den zukünftigen Lehrkräften den benötigten Gestaltungsspielraum, um Kompetenzen in ihrem Unterricht anzubahnen und zu erreichen. Gleichzeitig werden die Prozess- und Inhaltsbereiche knapp dargestellt und mit Beispielen veranschaulicht, um den Zugang zu diesem – für die Lehrkräfte an Grundschulen meist völlig fremden – Lerngegenstand zu ermöglichen.

Literatur

- [Fr16] Fricke, M.; Haselmeier, K.; Humbert, L.; Müller, D.; Rumm, P.: Informatik im Unterricht – So geht's, Plakatgrafik zu Informatik an Grundschulen, Wuppertal, 2016, URL: <http://metager.to/0t0x1>, Stand: 16.08.2017.
- [GI17] GI: Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für den Primarbereich, Erarbeitet vom Arbeitskreis „Bildungsstandards Informatik im Primarbereich“ – GI – Gesellschaft für Informatik (GI) e. V., 18. Juli 2017, URL: <http://metager.to/gibspdf>, Stand: 16.08.2017.
- [MS17] MSB NRW, Hrsg.: Informatik an Grundschulen, Pilotprojekt zur Erprobung von Konzepten zur informatischen Bildung im Rahmen des Sachunterrichts an Grundschulen, MSB NRW – Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen, Juli 2017, URL: <http://IaG.nrw.de/>, Stand: 16.08.2017.

Jugendwettbewerb Informatik: BWINF füllt die Lücke

Wolfgang Pohl, Robert Czechowski¹

Abstract: Durch Einführung des Jugendwettbewerbs Informatik (JwInf) mit begleitenden Lernangeboten zu Informatik-Grundlagen und Programmierung schließen die Bundesweiten Informatikwettbewerbe (BWINF) die Lücke zwischen Informatik-Biber und Bundeswettbewerb Informatik.

Keywords: Informatikwettbewerbe, Schülerwettbewerbe, Talentförderung

Das Projekt „Bundesweite Informatikwettbewerbe“ (BWINF) soll im Sinne der Vereinigung von Breiten- und Spitzenförderung pyramidal organisiert sein [Poh16], mit Wettbewerbsformaten, die einerseits aufeinander aufbauen, deren (potenzieller) Teilnahmekreis sich andererseits durch steigende fachliche Anforderungen nach und nach verkleinert. Bis 2016 gab es drei Stufen in dieser Wettbewerbspyramide:

- Auf der Eingangsstufe steht der Informatik-Biber [PSH09]. Dieses Online-Format wird seit 2007 jährlich angeboten, bis 2015 für die Sekundarstufe und seit 2016 auch für die Primarstufe (Jahrgangsstufen 3 und 4). In diesem Format können Teilnehmende ihre Neigung für das „digitale Denken“ (engl.: computational thinking) der Informatik entdecken und werden motiviert, sich näher mit der Disziplin auseinanderzusetzen. Am Informatik-Biber nahmen im Jahr 2016 in Deutschland über 290.000 Kinder und Jugendliche teil.
- Die zentrale Stufe bildet der Bundeswettbewerb Informatik (BwInf) [Poh06]. 1980 gegründet, wird er seit 1984 jährlich als Aufgabenwettbewerb durchgeführt. Ein Erfolg in diesem Leistungswettbewerb setzt erste Fachkenntnisse wie Wissen über grundlegende algorithmische Prinzipien und erste Programmierfertigkeiten voraus. Neben der fachlichen Vertiefung trägt der BwInf dazu bei, Jugendliche mit besonderem Potenzial zu erkennen, und ermöglicht den Teilnehmenden, ihre Begabung auszuloten.
- Die besten Teilnehmenden des BwInf haben schließlich die Chance, sich über ein Auswahlverfahren für das deutsche Team bei der Internationalen Informatikolympiade (IOI) zu qualifizieren. Diese dritte Wettbewerbsstufe fördert Jugendliche mit außerordentlichem Potenzial und vermittelt Fachkenntnisse auf Hochschulniveau.

Dieser Aufbau weist aber noch eine deutliche Lücke auf. Eine Teilnahme am Informatik-Biber ist ohne Fachkenntnisse und mit wenig Aufwand möglich. Wird durch die Biber-Teilnahme das Interesse an Informatik erfolgreich geweckt, müssen zur Teilnahme am BwInf erste Fachkenntnisse erworben werden; in der Schulbildung werden diese in der

¹ BWINF (Bundesweite Informatikwettbewerbe), Reuterstr. 159, 53113 Bonn, {pohl|czechowski}@bwinf.de

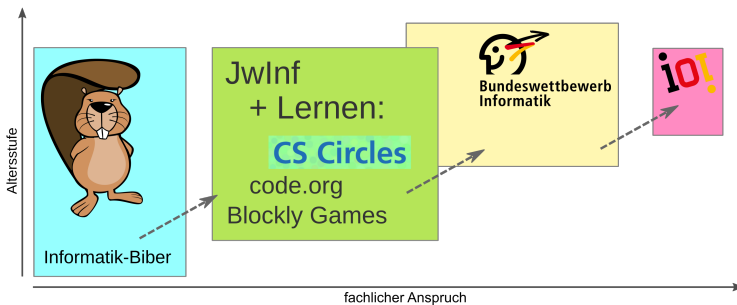


Abb. 1: Die „BWINF-Wettbewerbskarriere“

Regel erst in der Sekundarstufe 2 vermittelt. Außerdem ist die BwInf-Teilnahme mit hohem Aufwand verbunden. Um Biber-TeilnehmerInnen eine frühere und weniger arbeitsreiche Wettbewerbsteilnahme zu ermöglichen, führt BWINF ab 2017 ein neues Wettbewerbsangebot ein: den Jugendwettbewerb Informatik (JwInf). In der ersten Runde wird die Teilnahme in einer Online-Plattform realisiert; zu lösen sind in Mikrowelten eingebettete Probleme, in der Regel durch grafische Programmierung mit kontextbezogener Semantik. In der zweiten Runde sollen dann Aufgaben wie im BwInf zu Hause bearbeitet werden, mit Hilfe frei wählbarer Werkzeuge.

Auch hierzu sind allerdings erste fachliche Kompetenzen nötig oder zumindest hilfreich. Deshalb werden im Kontext des JwInf passende Lernangebote gemacht: etwa der seit einigen Jahren erfolgreich erprobte Online-Programmierkurs CS Circles² oder der Kurs „Grundlagen der Informatik“ von code.org³, an dessen Lokalisierung sich BWINF beteiligt hat. Mit dem neuen Wettbewerbsformat und den begleitenden Lernangeboten wird es möglich, die BWINF-Wettbewerbsformate fast nahtlos zu durchlaufen (vgl. Abb. 1).

Es wird zu beobachten sein, wie gut der Lückenschluss zwischen Informatik-Biber und Bundeswettbewerb Informatik gelingt. Wesentliches Maß für den Erfolg wird die Resonanz auf die neuen Angebote und mittelfristig auch die Entwicklung der Teilnahme am Bundeswettbewerb Informatik sein.

Literatur

- [Poh06] Wolfgang Pohl. Wettbewerb im Silberglanz. *LOG IN*, 26(141/142):10–13, 2006.
- [Poh16] Wolfgang Pohl. BWINF: Informatik mit Begeisterung entdecken. In Heinrich C. Mayr, Martin Pinzger (Hrsg.): *INFORMATIK 2016*, Seiten 1149–1152, Bonn, 2016. Gesellschaft für Informatik.
- [PSH09] Wolfgang Pohl, Kirsten Schlüter und Hans-Werner Hein. Informatik-Biber: Informatik-Einstieg und mehr. In Bernhard Koerber, Hrsg., *Zukunft braucht Herkunft: 25 Jahre INFOS – Informatik und Schule*, Seiten 38–49, Bonn, 2009. Gesellschaft für Informatik.

² cscircles.cemc.uwaterloo.ca/de

³ studio.code.org/s/20-hour

Zur Berechnung der Komplexität von einfachen objektorientierten Programmen

Marc Roßner¹

Abstract: Das Komplexitätsmaß nach Peter Rechenberg lässt sich auf einfache objektorientierte Programme übertragen. Solche Programme sind Gegenstand des Informatik-Unterrichts. Dies stellt eine Grundlage dafür dar, unterschiedliche Programme zu einer Aufgabe zu vergleichen.

Keywords: Komplexitätsmaß, Objektorientierung, Abituraufgaben

1 Übertragen der Software-Metrik von Rechenberg auf einfache objektorientierte Programme und C++

Unter einfachen objektorientierten Programmen sollen in dieser Arbeit solche Programme verstanden werden, die sich ausschließlich auf die Grundkonzepte Objekt und Klasse beziehen. Diese Konzepte bieten die Möglichkeit des strukturierten Aufbaus von Softwareprojekten. Man kann Teillösungen in Klassen auslagern und diese in einem Hauptprogramm zur Gesamtlösung vereinen. Bei diesem Vorgehen kann die Implementierung der Methoden als quasi-prozedural angesehen werden; die Methoden können in der Software-Metrik wie gewöhnliche Prozeduren behandelt werden. Für das Berechnen der Anweisungs- und der Ausdruckskomplexität können die Regelungen aus [Re86] unmittelbar übernommen werden. Das Berechnen der Datenkomplexität kann bei den Methoden so erfolgen, wie es bei Rechenberg für die modulare Programmierung beschrieben ist. Bei der Datenkomplexität werden Namen in Modulen als lokal und erst bei deren Verwendung im Hauptprogramm als global bewertet. Sollten Module ineinander geschachtelt sein, greift die Bewertung mit der Blocktiefendifferenz. Für Attribute soll gelten, dass sie im Sinne gekapselter Zustandsdaten, auf die alle oder fast alle Methoden der Klasse zugreifen, als global bewertet werden – unabhängig davon, ob innerhalb der Implementierung der Klasse oder im Hauptprogramm.

2 Analysieren der Aufgabe „Plumpsack“

Das Spiel „Plumpsack“ [Ab08] ist mithilfe einer Ringliste zu realisieren. Die Ringliste wird per Array implementiert. In Tabelle 1 sind die Bewertungen der vollständigen prozeduralen und objektorientierten Programme zum Spiel „Plumpsack“ angegeben.

¹ Friedrich-Schiller-Universität Jena, Fakultät für Mathematik und Informatik, Ernst-Abbe-Platz 2, 07743 Jena, marc.rossner@uni-jena.de

	prozedural	objektorientiert
Anweisungskomplexität	121.75	112.75
Ausdruckskomplexität	118	110
Datenkomplexität	122	136
Gesamtkomplexität	361.75	358.75

Tab. 1: Bewertung von Programmen zur Aufgabe „Plumpsack“.

Die Gesamtkomplexitäten der erarbeiteten Programme sind praktisch gleich. Daraus lässt sich ableiten, dass man es in der Abiturprüfung dem Prüfungsteilnehmer überlassen kann, ob er eine Aufgabe prozedural oder objektorientiert löst (bei Beschränkung auf die Grundkonzepte Klasse und Objekt).

3 Analysieren der Aufgabe „Liste“

Eine einfach verkettete Liste ist mit Hilfe eines Arrays zu realisieren [Ab07]. In Tabelle 2 sind die Bewertungen von drei unterschiedlichen Lösungsmöglichkeiten der Aufgabe angegeben.

	Zwei 1D- Arrays	Ein 2D- string-Array	Ein 1D- struct-Array
Anweisungskomplexität	339.5	377	338.5
Ausdruckskomplexität	200	258	221.5
Datenkomplexität	308	312	361
Gesamtkomplexität	847.5	947	921

Tab. 2: Bewertung von drei Musterlösungen zur Aufgabe „Liste“.

Wenn man dem Prüfungsteilnehmer die konkrete Realisierung der Datenstruktur überlässt, muss man damit rechnen, dass sich die Gesamtkomplexitäten der entwickelten Programme in der Größenordnung 10% voneinander unterscheiden. Diese Größenordnung erscheint auch in einer Abiturprüfung vertretbar. Nachdenklich stimmt der Unterschied in den Gesamtkomplexitäten von Programmen zu den beiden Abituraufgaben, obwohl mit beiden Aufgaben die gleiche Anzahl von Bewertungseinheiten erreichbar war (jeweils 30 BE von 60 BE für die gesamte Abiturprüfung).

Literaturverzeichnis

- [Ab07] Freistaat Thüringen: Abiturprüfung 2007 Leistungsfach Informatik (Haupttermin).
- [Ab08] Freistaat Thüringen: Abiturprüfung 2008 Leistungsfach Informatik (Haupttermin).
- [Re86] Rechenberg, Peter: Ein neues Maß für die softwaretechnische Komplexität von Programmen. In: Informatik Forschung und Entwicklung (1986) 1: 26-37.

Wie stellen sich Schülerinnen und Schüler informatische Konzepte vor und welche sprachlichen Bilder nutzen sie?

Arne Saathoff¹

Abstract: Viele Jugendliche wissen nicht, wie grundlegende Technologien der Informatik bzw. bestimmte informatische Konzepte funktionieren. Die Kompetenzen jüngerer SuS² beschränken sich oftmals auf das Benutzen bestimmter Hard- und Softwares, wodurch sie lediglich Produktwissen erlangen. Erstrebenswerter ist es aber, den SuS ebenfalls Konzeptwissen zu vermitteln, damit sie in der Lage sind, Gelerntes in einen größeren Zusammenhang einzuordnen und früher erworbene Kompetenzen auf neue Situationen anzuwenden. In dieser Untersuchung werden Interviews mit SuS analysiert, um Kenntnisse darüber zu erlangen, wie sich SuS informatische Inhalte erschließen und welche Formen des Relational Reasoning dabei von ihnen genutzt werden.

Keywords: Relational Reasoning, Analogie, Anomalie, Antinomie, Antithese, Metapher

1 Einleitung

In der durchgeführten Studie sollen die Vorkenntnisse und Vorstellungen von SuS bezüglich informatischer Sachverhalte, die insbesondere alltägliche Aktivitäten im Internet betreffen, untersucht werden. Das Erkenntnisinteresse liegt darin, die von SuS verwendeten Formen des Relational Reasoning zu ermitteln, damit Rückschlüsse auf den Gebrauch von sprachlichen Konstrukten, wie Vergleiche mit bereits bekannten Sachverhalten oder die konkrete Abgrenzung von unzutreffenden Konzepten, gezogen werden können. Jablansky, Alexander, Dumas und Compton [JADC15] hypothetisierten im Jahr 2015, dass die Analyse technologischer Objekte, vor allem solcher, die neuartig und unbekannt sind, Kindern und Jugendlichen die Möglichkeit bietet, vergleichend zu begründen („to reason relationally“ [JADC15], p. 3). Ein Teil des Ergebnisses war die Erkenntnis, dass alle befragten SuS in irgendeiner Form das Relational Reasoning anwenden, wobei die Analogie die am häufigsten auftretende Form war. Analogien zeichnen sich durch das Heranziehen von Vergleichen aus. Gerade deshalb können sie genutzt werden, um komplexe Sachverhalte mithilfe von analogen Sprachbildern (Metaphern) zu verdeutlichen. Besonderes Augenmerk liegt daher auf der Frage, welche Rolle Metaphern dabei spielen könnten, informatische Inhalte zu versinnbildlichen. Somit dient diese Ausarbeitung einerseits der Forschung im Bereich der Didaktik der Informatik, andererseits soll sie Lehrkräften einen Einblick in die Vorstellungen von SuS geben und diesbezüglich deren Verwendung von Sprache aufzeigen.

¹ Universität Oldenburg, Department für Informatik, Abteilung Informatik in der Bildung, 26111 Oldenburg, arne.saathoff@uni-oldenburg.de

² Im Folgenden als Abkürzung für Schülerinnen und Schüler

2 Auswertung der Interviews

Das Datenmaterial dieser Untersuchung entstammt einer im Jahr 2010 verfassten Masterarbeit. Dabei wurden 11 Interviews mit insgesamt 23 SuS geführt. Die Interviewpartner konnten damals durch eine Kooperation mit zwei Schulen gewonnen werden. Daher handelt es sich um eine Sekundäranalyse, die mithilfe der qualitativen Inhaltsanalyse nach Mayring [May15] durchgeführt wurde. Dabei fiel auf, dass die SuS fast ausschließlich die Analogie heranzogen, indem sie einen unbekanntem Sachverhalt bzw. eine unbekanntete Funktionsweise des Internets mit einem ihnen vertrauten Sachverhalt verglichen. Im Zuge dieses Vergleichens wurde in allen Fällen ein sprachliches Bild (Metapher) verwendet. Um einen Einblick in die vollständige Auswertung und die Datensätze zu erhalten, sei auf die Masterarbeit [Saa16] verwiesen.

3 Ergebnisse und Fazit

Analogien, wie die Männchen- bzw. Post-Metapher, sind gut geeignet, um die allgemeinen Vorgänge und Abläufe im Internet zu versinnbildlichen. Einzelne Phänomene, wie Streaming, können durch den Vergleich mit dem Kurzzeitgedächtnis oder einem Kopierer erklärt werden. Die Geschwindigkeit der Datenübertragung wird mithilfe von Multitasking oder dem Gespräch beschrieben. Die Klötzchen-Metapher dient dazu, die Paketierung im Zuge der Datenübertragung zu veranschaulichen. Lehrkräfte könnten die genannten Analogien als Grundlage für die Unterrichtsplanung nutzen. Dabei eignen sich Metaphern insbesondere zur bildlichen Darstellung von komplexen informatischen Sachverhalten, die zum ersten Mal behandelt werden. Die Komplexität eines informatischen Konzepts kann durch ein analoges Konzept, welches SuS vertraut ist, reduziert werden. Auf Grundlage dieser Versinnbildlichung muss dann wiederum der Bezug zu der tatsächlichen wissenschaftlichen Sichtweise hergestellt werden. Dieser Vorgang birgt einerseits Gefahren, wenn sich SuS womöglich nicht ausreichend von einer Metapher distanzieren können. Andererseits birgt der Einsatz geeigneter Metaphern durchaus Potenzial, SuS eine Hilfestellung bzw. ein Gedankengerüst zu bieten, um informatische Konzepte zu durchdringen.

Literaturverzeichnis

- [JADC15] Jablansky, S.; Alexander, P. A.; Dumas, D. & Compton, V. (2015). Developmental Differences in Relational Reasoning Among Primary and Secondary School Students. *Journal of Educational Psychology*, 108(4), pp. 592–608.
- [May15] Mayring, P. (2015). *Qualitative Inhaltsanalyse - Grundlagen und Techniken* (12. Aufl.). Weinheim und Basel: Beltz Verlag.
- [Saa16] Saathoff, A. (2016). *Analyse der Verwendung von Relational Reasoning in Schülerinterviews über informatische Konzepte*. Masterarbeit. Universität Oldenburg.

Sei selbst der Igel! – Umsetzung ausgewählter Elemente der phänomenorientierten Informatik

Daniel Siebrecht¹

Abstract: In diesem Beitrag wird ein Konzept vorgestellt, das Bewegung bewusst als informatischen Unterrichtsgegenstand auffasst und ihn mit der Algorithmik verknüpft. Dadurch bieten sich nicht nur die klassischen Formen der symbolischen oder ikonischen Visualisierung von Problemstellungen an: Naheliegender kann ebenso ein enaktiver Zugang gewonnen werden. Die Lernenden führen im vorliegenden Konzept die Algorithmen selbst durch, sodass sich schließlich das Ausführen algorithmisierter Bewegung als ergänzende Alternative zur gewohnten Implementation am Informatiksystem herausstellt.

Keywords: Implementation; Bewegung; Algorithmik; enaktiv

1 Bewegung als informatischer Unterrichtsgegenstand

Üblicherweise wird Bewegung im Zusammenhang mit Informatikunterricht *methodisch* assoziiert. Namhafte Arbeiten in der internationalen Informatikdidaktik legen nahe, dass sich die enaktive Erarbeitung informatischer Konzepte in Form größerer Motivation und ergiebigeren Lernerfolgs rentiert (z. B. [BWF15]). Rollenspiele und Aktivitäten sind eindeutig einer *methodischen* Aufwertung des Informatikunterrichts zuzurechnen. Ein Informatikunterricht, der aber beispielsweise die Straßenüberquerung oder den „Tanz der Bienen“ analysiert und modelliert, versteht Bewegung als *Sachgegenstand*.

Als wesentliche Bezugsthematik ist die Algorithmik der Bewegung zuzuordnen: Bei der Notation von Bewegungsabläufen (Kinetographie) werden algorithmische Strukturen verwendet. Verkehrsnetze – etwa auch Labyrinth – bergen vornehmlich „Weg-Problemstellungen“. Hüpfkästchen und schrittmusterartiges Bewegungen oder gar Tänze bieten hingegen keine direkte Problemstellung, setzen sich aber aus Strukturen algorithmischer Elemente zusammen. Nicht kinetisch bedingte Algorithmen wären hingegen beispielsweise der Euklid-Algorithmus, String-Matching-Algorithmen oder Algorithmen zur Datenbankanalyse. Somit lässt sich eine Schnittmenge zwischen Algorithmik und Bewegung (Kinalgorithmik) finden, „die sich mit der Untersuchung, Lösung und Umsetzung von kinetographischen Problemstellungen auseinandersetzt“ ([Si15, S. 27]). Zieldimension ist das Analysieren und Modellieren kinetischer Strukturen und Probleme, deren Lösung perspektivisch zu einer Ausführung gebracht werden soll, sei es nun auf einem Informatiksystem oder durch einen Menschen.

¹ Bergische Universität Wuppertal, Didaktik der Informatik – siebrecht@uni-wuppertal.de

2 Alternative Implementation durch Bewegung

Der Vorschlag ist also eine Alternative zur „klassischen“ Implementation am Informatiksystem in Form menschlicher Bewegungsausführung (anthropokinetische Implementationsalternative). Dabei mögen virtuelle Systeme, die vornehmlich den visuellen Lernkanal bedienen, in den Hintergrund treten, um durch Haptik und Motorik enaktiv Lernprozesse zu initiieren und zu fördern, welche zur Aneignung informatischer Kompetenzen auf *konzeptueller* Ebene befähigen sollen. Statt einen Editor zu öffnen, um Quelltext zu verfassen, und diesen anschließend kompilieren zu *lassen* oder interpretieren zu *lassen*, böte sich den Schülerinnen und Schülern die Möglichkeit, nach einer zeiteffizienten Notation der Programmsequenz mit Stift und Papier dieses Programm unmittelbar im Anschluss an die Entwicklung *selbst* als Programm ausführende Instanz zu übersetzen und zwar in eigene Körperbewegungen. Dies kann aber nur umgesetzt werden, wenn der modellierte Gegenstand kinalgorithmischer Natur ist. „Die eigentliche Sprache, die es dabei zu lernen gilt, ist [nämlich] weniger eine beliebige Programmiersprache, sondern eher die Algorithmik selbst“ ([Si15, S. 28]). DISSMANN fordert in diesem Sinne eine zielgerichtete Handlungsfähigkeit „im Umgang mit den programmtechnischen Konzepten“ ([Di03, S. 226]); der „Einsatz der natürlichen Fähigkeiten“ ([Di03, S. 228]) sei dabei besonders wichtig; er verweist darauf, dass die Lernenden bei einer Informatiksystem gestützten Implementation mit dem „trial and error“-Prinzip auf die Rückmeldungen des Compilers antworten. Bei der angedachten Alternative zeigt sich ein eindeutiger Vorteil darin, dass die Lernenden dazu angehalten wären, sich *selbst* hinsichtlich einer fehlerbehaftet notierten Programmsequenz korrigieren zu müssen – ohne dabei die Schuld auf „Dritte“ schieben zu können.

Literatur

- [BWF15] Bell, T.; Witten, I. H.; Fellows, M.: Teaching Computer Science Unplugged. Teachers' Edition. An enrichment and extension programme for primary-aged students. Computer Science Unplugged, 2015.
- [Di03] Dißmann, S.: Handlungsorientiertes Erlernen von Programmkonstruktionen anhand von Rollenspielen. In (Hubwieser, P., Hrsg.): Informatische Fachkonzepte im Unterricht, INFOS 2003, 10. GI-Fachtagung Informatik und Schule, 17.-19. September 2003 in Garching bei München. Bd. 32. Lecture Notes in Informatics, Gesellschaft für Informatik, Bonn, S. 225–235, 2003.
- [Si15] Siebrecht, D.: Einführung algorithmischer Elemente unter kinetographischen Aspekten, 2015, URL: <http://bscw.ham.nw.schule.de/pub/bscw.cgi/6586882>, Stand: 17.08.2017.

Ergänzendes eLearning-Angebot für pädagogische Fachkräfte aus dem Vor- und Grundschulbereich zum leichten Einstieg in die Elementarinformatik

Anke Steinhäuser¹

Abstract: Um informatische Ansätze bereits in der Vor- und Grundschule zu etablieren, müssen gezielte eLearning-Konzepte entwickelt werden, die auch bei weniger medienaffinen pädagogischen Fachkräften Freude und Interesse hervorrufen. Exemplarisch wird eLEx vorgestellt, ein eLearning-Angebot zum leichten Einstieg in die Elementarinformatik. Die digitalen Lerneinheiten bilden ein eigenes Konzept, das sich aus der Experimentierkiste Informatik [Sc16] begründet und dabei helfen soll, bei Lehrenden Synergien zwischen Elementarinformatik und Medienbildung zu erzeugen.

Keywords: Elementarinformatik, digitale Lerneinheiten, Medienbildung

1 Einleitung und Motivation

Die digitale Bildungsoffensive ist eine große Herausforderung für alle Lehrkräfte, die weiterhin im „Anlogen“ verhaftet sind und digitalen Technologien skeptisch gegenüberstehen [In14]. Um auch die Gruppe von medien- und informatikfernen Pädagoginnen und Pädagogen für die Grundlagen der Informatik zu begeistern, braucht es gezielte motivationale Schulungskonzepte, um Ängste abzubauen, Verständnis zu fördern und praxisorientierte Grundlagen [Re05] zu vermitteln. Daraus entstand die Idee von eLEx, einem e-Learning-Angebot, aufbauend auf der Handreichung zur Experimentierkiste Informatik [Sc16] unter Berücksichtigung multimedialer, mediendidaktischer und motivationaler Grundlagen.

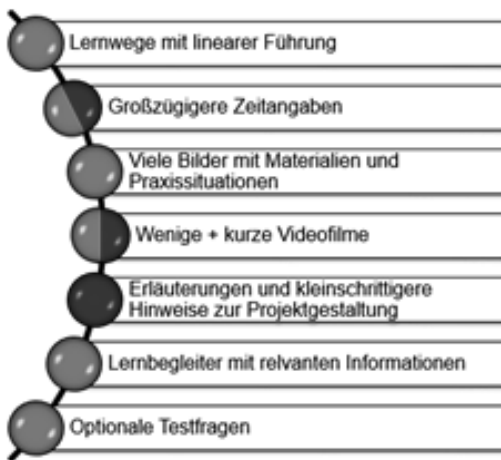
2 Projektziele und Vorgehen

Die digitalen Lerneinheiten von eLEx haben hohen Praxisbezug und sollen Lehrkräfte zur reflexiven Auseinandersetzung anregen. Ziel ist es, Synergien zwischen digitaler Medienbildung und Einstellungsänderungen zur Informatik zu erzeugen. Mit eLEx wurde deshalb ein leicht zugängliches und anschauliches Fortbildungsangebot als Vorbereitung für den Einsatz der Experimentierkiste Informatik konzipiert.

¹Otto-Friedrich-Universität Bamberg, Fakultät Wirtschaftsinformatik und Angewandte Informatik, Forschungsgruppe Elementarinformatik, An der Weberei 5, 96045 Bamberg, ute.schmid@uni-bamberg.de, zgl. Friedrich-Alexander-Universität, Erlangen-Nürnberg, Philosophische Fakultät, Master Multimedia Didaktik, eLEx unveröffentlicht, erhältlich via E-Mail von anke-steinhaeuser@gmx.de.

3 Empirische Studie und Ergebnisse

Die qualitative Erhebungsmethode wurde mit einem teilstrukturierten Interviewleitfaden und der Auswertung durch MAXQDA an fünf Testpersonen durchgeführt. Produkt- und lernprozessbezogene Fragen zu Usability, Selbstwirksamkeit, Kompetenzerleben und die Teilnahme an weiteren ähnlichen Schulungen standen im Fokus. Die Zielgruppe schätzte die Darbietung praxisnaher Beispiele für ihren pädagogischen Kontext als relevant und praxistauglich ein, hatte Spaß an eLex und traute sich infolgedessen auch den Einsatz der Experimentierkiste Informatik in der eigenen Einrichtung zu.



Die Abb. zeigt die Bedürfnisse der Zielgruppe in Bezug auf eLex: Die helle Füllung wird als positiv, die zweifarbige als heterogen und die dunkle als verbesserungswürdig eingestuft. Der Wunsch nach kleinschrittigeren Anleitungen wurde mehrfach geäußert. Insgesamt wurde eLex als bereichernd empfunden. Zudem wurde vorgeschlagen, eLex in der Arbeit mit jungen Flüchtlingen einzusetzen und die „Kiste“ modifiziert für fünfte und sechste Klassen anzubieten.

4 Diskussion und Ausblick

Medienbildung verknüpft mit informatischen Grundlagen fördern ein nachhaltiges Verständnis für Informatik in der digitalen Welt. Lehrkräfte aus dem Vor- und Grundschulbereich ohne informatische und mit nur geringen digitalen Vorkenntnissen werden mit eLex stärker in den Prozess miteinbezogen [Th16]. Weitere Studien sind in Planung.

Literaturverzeichnis

- [In14] Initiative D21 (2014). Medienbildung an deutschen Schulen. Handlungsempfehlungen für die digitale Gesellschaft. URL: www.initiatived21.de (Stand 02.06.2017).
- [Sc16] Schmid, U., Weitz, K., Wolking, M. (2016). Handreichung zur Experimentierkiste Informatik, Otto-Friedrich-Universität Bamberg, unveröffentlicht, erhältlich via E-Mail von ute.schmid@uni-bamberg.de.
- [Th16] Thomas, M., Weigend, M. (Hrsg.) (2016). Informatik für Kinder. 7. Münsteraner Workshop zur Schulinformatik – 20. Mai 2016. Universität Münster: BoD, S.113-119.

Informatik in der Elementarpädagogik – Herausforderung für die Professionalisierung pädagogischer Fachkräfte

Maarit Stierle¹

Abstract: Die Realisierung und der Erfolg informatischer Frühförderangebote in Kindertagesstätten sind hauptsächlich von der Qualifizierung der pädagogischen Fachkräfte abhängig. Dieser Beitrag betont notwendige Maßnahmen im Rahmen der Professionalisierung frühpädagogischer Fachkräfte. Abschließend führt er weitere relevante Vorarbeiten und Forschungsdesiderate für die Implementierung des Bildungsbereichs Informatik an.

Keywords: Elementarpädagogik, Professionalisierung, pädagogische Fachkräfte, informatische Frühförderung, Kindertagesstätte, MINT, Informatikdidaktik in der beruflichen Bildung

1 Professionalisierung pädagogischer Fachkräfte für die informatische Frühförderung

MINT-Fächer sind für viele ErzieherInnen mit negativen Emotionen besetzt. Nicht selten entschieden sie sich mit ihrer Ausbildung bewusst gegen einen Beruf mit naturwissenschaftlichen und technischen Inhalten. Es müssen daher dringend Maßnahmen ergriffen werden, um Vorurteile und Ängste zu überwinden sowie das generelle Interesse an MINT-Themen zu stimulieren. Angehenden pädagogischen Fachkräften ist die Angst zu nehmen, dass informatische Erziehung zu komplex für sie sei. Außerdem sollten sie die informatische Frühförderung als Bestandteil ihres Berufsbilds erkennen. Mit ersten Workshops im Rahmen der Forschungsgruppe Elementarinformatik konnten angehende ErzieherInnen an einer Fachakademie für Sozialpädagogik bereits für die informatische Frühförderung sensibilisiert werden. Die ErzieherInnen zeigten sich anfangs zwar noch skeptisch, verstanden aber nach kurzer Zeit, warum Informatik schon im Kindergartenalter eine wichtige Rolle spielt. Sie entwickelten nach einer thematischen Einführung sogar eigene Ideen zur Umsetzung in der Praxis.

Die Integration informatischer Frühförderung in die schulische Ausbildung der pädagogischen Fachkräfte ist dringend erforderlich. Sinnvoll erscheint es den MNE durch die Etablierung eines MINT-Unterrichts abzulösen, damit auch die technische Bildung in die beruflichen Schulen Einzug hält und die Naturwissenschaftliche Erziehung den Anforderungen des digitalen 21. Jahrhunderts angepasst wird. Der Unterricht sollte didaktisch so gestaltet werden, wie die Förderung auch tatsächlich in

¹ Universität Bamberg/Forschungsgruppe Elementarinformatik (FELI), An der Weberei 5, 96045 Bamberg, maarit.stierle@gmail.com

den frühkindlichen Institutionen für Kinder durchgeführt wird, die weder lesen noch schreiben können. Dementsprechend handelt es sich um eine kindgerechte Unterrichtsgestaltung, sodass den Auszubildenden didaktische und methodische Kenntnisse für die Praxis vermittelt werden. Darüber hinaus benötigen die SchülerInnen und Studierenden auch Hintergrundwissen. Grundprinzipien für den MINT-Unterricht als auch später für die informatische Erziehung in den Kindertagesstätten sind handlungsorientiertes, problemlösendes und entdeckendes Lernen. Die Auszubildenden können sich im Unterricht in ihre späteren Kindergartenkinder hineinversetzen und dabei u.a. mögliche aufkommende Fragen entwickeln, diese stellen und darauf angemessene Antworten oder Reaktionen finden. Die Lehrkraft nimmt eine tendenziell begleitende Rolle im MINT-Unterricht ein und hält sich im Hintergrund des Geschehens. Die Auszubildenden dürfen möglichst viel selbst testen und dabei ihre kommunikativen Kompetenzen ausbilden, um in ihrem späteren Beruf ko-konstruktive Lernprozesse in den Kindertagesstätten anregen zu können.

Die höhere Gewichtung der informatischen Frühförderung in der Ausbildung pädagogischer Fachkräfte setzt entsprechende Kenntnisse bei den Lehrkräften für berufliche Schulen voraus. Der doppelte pädagogische Bezug der Lehrkräfte erfordert, dass diese über fachliche sowie fachdidaktische Kompetenzen verfügen und den Auszubildenden praktisch vermitteln können, wie informatische Frühförderung in der Praxis stattfinden kann. Auch hier gilt es, die MINT-Hochschuldidaktik in der LehrerInnen-Ausbildung an die Umsetzung in den Kindertagesstätten anzupassen.

2 Ausblick

Die informatische Frühförderung wird in den nächsten Jahren verstärkt in den Fokus der Bildungsdiskussionen rücken. Für die Realisierung der Informatik als Bildungsbereich in Kindertagesstätten sind Maßnahmen bei der Ausbildung pädagogischer Fachkräfte notwendig. Daneben stehen weitere Aufgaben für die erfolgreiche Implementierung informatischer Frühförderung aus: die Festlegung von Bildungsstandards und die Überarbeitung der Bildungspläne der Länder, die Entwicklung konkreter Materialien zur Frühförderung sowie Handreichungen für pädagogische Fachkräfte, Ausbau von Fort- und Weiterbildungsangeboten, die Erarbeitung von Instrumenten zur Messung informatischer Kompetenzen und der Effektivität von Förderangeboten, Untersuchungen, wie sich Angebote zur Förderung informatischer Kompetenzen in bestehende pädagogische Konzepte von Kindertagesstätten integrieren lassen und viele weitere Handlungsschritte, die sich sowohl direkt als auch indirekt auf die Bildungsqualität der Kindertagesstätten auswirken.

Förderung digitaler Bildung mit digitalen Schulbüchern

Manuel Froitzheim¹, Klaus Becker², Martin Zimmol³, Daniel Jonietz⁴, Oliver Schneider⁵,
Heiko Jochum⁶

Abstract: Das elektronische Schulbuch inf-schule.de ist ein interaktives und multimediales Schulbuch für den Informatikunterricht. Im folgenden Beitrag werden die Idee des elektronischen Schulbuchs, sowie die Förderung der informatischen Bildung durch inf-schule.de dargestellt und mit Hilfe einer umfangreichen Evaluierung belegt. Die Befragung per Fragebogen von 1170 Lernenden und 175 Lehrenden gibt zum Beispiel Aufschluss über die Möglichkeiten, die für den Informatikunterricht durch den Einsatz des Schulbuchs entstehen.

Keywords: Elektronisches Schulbuch, Digitales Schulbuch, Informatikunterricht, Lernen, Didaktik der Informatik, Web-Applikation

1 Einleitung und Motivation

Digitale Bildung wird inzwischen als unverzichtbar in einer immer stärker digital geprägten und vernetzten Welt angesehen. Die Aneignung grundlegender Konzepte zum Verständnis dieser Welt und die Entwicklung von Kompetenzen zur Mitgestaltung einer solchen Welt sollte in einem eigenständigen Lernbereich erfolgen – so die einhellige Forderung aller mit digitaler Bildung befassten Fachdidaktiker [Da16]. Nur in einem fundierten Informatikunterricht kann die anwendungsbezogene Perspektive (Wie nutze ich das?) und die gesellschaftlich-kulturelle Perspektive (Wie wirkt das?) um die zum Verständnis der digitalen Welt erforderliche technologische Perspektive (Wie funktioniert das?) erweitert werden.

Der Erwerb digitaler Bildung wird im Informatikunterricht gezielt gefördert. Das Schulbuch als Leitmedium bei der Unterrichtsplanung und Unterrichtsgestaltung spielt dabei eine zentrale Rolle [Wi03], weil hier „Lerninhalte eines Faches oder Lernbereichs in systematischer, didaktischer und methodischer Form“ bereitgestellt werden [Sa10].

Schulbücher wurden bisher in Buchform erstellt. Mit der Entwicklung digitaler bzw.

¹ Universität Siegen, Zentrum für ökonomische Bildung in Siegen (ZöBiS), Kohlbettstraße 15, 57068 Siegen, froitzheim@zoebis.de

² BurgGymnasium Kaiserslautern, Burgstraße 18, 67659 Kaiserslautern, beckerkp@web.de

³ Pädagogisches Landesinstitut Rheinland-Pfalz, Butenschönstraße 2, 67346 Speyer, martin.zimmol@pl.rlp.de

⁴ BurgGymnasium Kaiserslautern, Burgstraße 18, 67659 Kaiserslautern, daniel@jonietz.de

⁵ Sickingen-Gymnasium Landstuhl, Philipp-Fauth-Straße 3, 66849 Landstuhl, o.schneider@mac.com

⁶ Werner-Heisenberg-Gymnasium Bad Dürkheim, Kanalstr. 19, 67098 Bad Dürkheim, jom@whgonline.de

elektronischer Schulbücher ist ein neues Forschungsfeld entstanden, das der Entwicklung von Lernumgebungen neue Impulse verleiht [ScFr14][ScFr15][ScFr16]. Gerade für die Förderung digitaler Bildung ergeben sich interessante Ansatzpunkte durch das Lernen mit einem digitalen Schulbuch – was im Folgenden genauer aufgezeigt werden soll.

Im Zentrum unseres Forschungsansatzes steht unter anderem die Frage, welche Möglichkeiten ein elektronisches Schulbuch bietet, um den Erwerb digitaler Bildung im Informatikunterricht noch effektiver und adressatengerechter zu gestalten, als dieses mit gedruckten Schulbüchern erfolgen kann.

Um dieser Frage nachzugehen, wurde und wird nach der Methode der fachdidaktischen Entwicklungsforschung [Pr12a][Pr12b] ein Schulbuch entwickelt und im Unterricht breit eingesetzt und getestet. Die Akzeptanz bei Lernenden und Lehrenden wurde durch eine Erhebung quantitativ und qualitativ evaluiert. [Fr16a][Fr16b] Im Folgenden wird der aktuelle Entwicklungsstand unter der Perspektive „Möglichkeiten eines digitalen Schulbuchs zur Förderung digitaler Bildung“ aufgezeigt.

2 Das Schulbuchprojekt inf-schule.de

Das Schulbuchprojekt inf-schule.de (vgl. <https://www.inf-schule.de/>) wurde 2008 initiiert und inzwischen ist ein elektronisches Schulbuch für den Informatikunterricht entwickelt worden, das fast die gesamten Unterrichtsinhalte des Faches Informatik abdeckt. Bezüglich des Umfangs (mit mehr als 2000 Webseiten) und der Nutzerzahlen (mit mehr als 100.000 Zugriffen pro Schulwoche) hat sich inf-schule.de inzwischen zum akzeptierten Lehrwerk für den Informatikunterricht entwickelt. Genau wie gedruckte Schulbücher orientiert es sich an den Lehrplänen und den Bildungsstandards und bietet vielfältiges Material zum Lernen und Lehren.

Das Schulbuch inf-schule.de wird als digitales Schulbuch im Internet angeboten. In der nach Sanguo, Xuehai und Chenglin [Sa12] vorgeschlagenen Entwicklungshierarchie für digitale Schulbücher ist es der Stufe 2.0 (Schulbücher ohne analoge Vorlage) zuzurechnen. In weiten Teilen erreicht es inzwischen auch die Stufe 3.0, die eine Mitgestaltung durch die Nutzer vorsieht.

3 Forschungsdesign

Mit einer breit angelegten Online-Fragebogenaktion wurde die Nutzung und Akzeptanz des digitalen Schulbuchs inf-schule.de bei Lernenden und Lehrenden ermittelt und unter wesentlichen Kriterien quantitativ und qualitativ evaluiert. Die hohen Nutzerzahlen ermöglichen es, über einen Online-Fragebogen zum einen eine ausreichend große Rückmeldung zu erhalten und zum anderen mit den vorhandenen Ressourcen möglichst viele Lehrende und Lernende zu erreichen, um die Akzeptanz und die Nutzung bezüglich des elektronischen Schulbuchs bei einer möglichst breiten Zielgruppe zu messen. Ein weiteres Ziel der Evaluierung bestand in der Erfassung von

Verbesserungsmöglichkeiten und Anregungen für die weitere Entwicklung, um in den folgenden Entwicklungszyklen entsprechende Verbesserungen in das elektronische Schulbuch zu integrieren.

An der Befragung haben sich 242 Schülerinnen (20,2%) und 928 Schüler (77,5%) beteiligt. Der geringe Anteil an Schülerinnen ist auf die Geschlechterverteilung in Informatikkursen zurückzuführen. Der Wert von 20,2% entspricht ungefähr dem Anteil der Frauen in den neuen IT-Berufen [HP10][BB15]. Die Lernenden sind durchschnittlich 16,9 Jahre alt. Der größte Teil der Lernenden ist der Sekundarstufe II (907 Lernende) zuzuordnen. In der Sekundarstufe I sind 150 Lernende, und aus dem tertiären Bildungsbereich (z.B. Universität und Fachhochschule) haben sich nur 12 Lernende an der Evaluierung beteiligt. An der Evaluierung haben zudem 175 Lehrende teilgenommen. Davon sind 29% weiblich und 71% männlich. Alle im Schuldienst vertretenen Altersgruppen haben sich an der Evaluierung beteiligt.

4 Möglichkeiten eines digitalen Schulbuchs

Attraktive Lernangebote sind Ausgangspunkt für eine gezielte Förderung digitaler Bildung. Im Folgenden werden die im digitalen Schulbuch inf-schule.de bereits umgesetzten Möglichkeiten eines digitalen Schulbuchs zur Gestaltung solcher Lernumgebungen aufgezeigt und die Ergebnisse der Evaluierung dargestellt.

4.1 Verfügbarkeit

Das digitale Schulbuch inf-schule.de wird frei verfügbar als Webseite im Internet bereitgestellt. Das Schulbuch kann daher von jedem Interessierten jederzeit an (fast) jedem Ort ohne irgendwelche Zugangsbarrieren genutzt werden. Der Aspekt der universellen und freien Verfügbarkeit im Internet wurde in der Evaluation mit Abstand am höchsten bewertet. Für 45% der befragten Personen (Lehrende und Lernende) hat dieser Aspekt die größte Bedeutung am Erfolg des elektronischen Schulbuchs. Hier spiegelt sich eine Entwicklung weg von Printmedien hin zu digitalen Medien wider, die insbesondere bei Jugendlichen zu beobachten ist [JIM15]. Wichtig ist es Jugendlichen auch, dass die dargestellte Information auf verschiedenen Endgeräten, vom Desktop-Computer bis zum Smartphone, flexibel angezeigt werden können. Die Personen nutzen das elektronische Schulbuch zu 35,6% auf einem Computer, zu 43,3% auf einem Laptop, zu 13,2% auf einem Tablet und zu 7,9% auf einem Smartphone. Im Fragebogen war die Auswahl mehrerer Endgeräte möglich und der hohe Anteil an Desktop-Computern lässt sich durch die genutzte Ausstattung während des Unterrichts erklären. Die Nutzung auf mobilen Endgeräten (21,1%) führt zur Notwendigkeit eines responsiven Designs des elektronischen Schulbuchs. Die befragten Personen äußerten eine starke Zustimmung oder Zustimmung bei der Darstellung auf Computern (97,9%) und Laptops (97,2%). Bei Tablets ist die Zustimmung zur guten Darstellung mit 74,3% etwas schlechter und bei Smartphones liegt die Zustimmung bei 57,4%. Eine entsprechende Optimierung der Darstellung auf mobilen Endgeräten ist in der Zukunft wünschenswert.

Mit der bisherigen Verfügbarkeit und Darstellung der Informationen im elektronischen Schulbuch gelingt es, Lernende und Lehrende anzusprechen und ihnen in zufriedenstellender Art Lernangebote zur digitalen Bildung zu unterbreiten.

Die freie Verfügbarkeit im Internet hat zudem zur Folge, dass nicht nur die ursprünglich vorgesehene Lerngruppe „Schülerinnen und Schüler“ angesprochen wird, sondern auch andere Interessierte. Aus vielfältigen Rückmeldungen, auch außerhalb der gezogenen Stichprobe, geht hervor, dass die Materialien zur Studienvorbereitung empfohlen werden oder unterstützend im Studium verwendet werden. Diese zunächst unbeabsichtigte Ausweitung des Nutzerkreises ist aber ganz im Sinne einer breit angelegten Vermittlung digitaler Bildung.

4.2 Digitale Darstellung der Inhalte

Die digitale Darstellung von Materialien erlaubt es, Lerninhalte in ihrer „natürlichen Form“ vorzugeben. Dies soll anhand eines Lernszenarios erläutert werden. Viele Fachkonzepte der Informatik lassen sich effektiv vermitteln, wenn geeignete Lernszenarien (z.B. als Programmfragment oder als ausführbares Modell) zum Analysieren, Variieren und Weiterentwickeln bereitgestellt werden. Nach einer einführenden Aufgabe, die den Sinn von eigenständigen Unterprogrammeinheiten motiviert, wird dem Lernenden ein ausführbares Programmfragment vorgegeben, das aufzeigt, wie die neue Idee umgesetzt werden kann. Dieses Programmfragment soll von den Lernenden analysiert und per Analogiebildung vervollständigt werden. (vgl. Abb. 1)

```
# Unterprogramme

def kleeblattreiheBisBaumLegen():
    while not kara.treeFront():
        kara.putLeaf()
        kara.move()
        kara.putLeaf()

...

# Hauptprogramm
kleeblattreiheBisBaumLegen()
umBaumLaufen()

...
```

Abb. 1: Beispiel: Einführung von Unterprogrammen (vgl. http://www.inf-schule.de/programmierung/kara/unterprogramme/erkundung_unterprogramm)

Aufwendig ist das Abtippen des Programmfragmentes, wenn es in einem Buch abgedruckt ist. Natürlicher ist es, wenn man direkt auf das Lernmaterial zugreifen kann. In einem digitalen Schulbuch lässt sich dies bei Quelltexten per „Copy and Paste“ erledigen, bei anderen vorgegebenen Materialien, indem man einem Link folgt.

Gerade bei der Vermittlung digitaler Bildung ist es günstig, wenn Lernmaterialien zu den digital geprägten Lerninhalten in digitaler Form dargestellt und verfügbar gemacht werden. Ein digitales Schulbuch ist von seiner Konzeption hierfür prädestiniert. Zudem unterstützt es die digitale Aufbereitung von Lernprodukten, indem das Schulbuch als Materiallieferant für ein digitales Heft oder digitales Portfolio fungiert. Durch die digitale Verfügbarkeit der Materialien kann die Unterrichtszeit für die digitale Bildung genutzt werden und muss nicht zum Abtippen von Quellcode verschwendet werden.

4.3 Multimediale Darstellung der Inhalte

Ein elektronisches Schulbuch kann neben der text- und bilderbasierten Darstellung weitere Darstellungsmöglichkeiten nutzen. Mit Videos, Animationen usw. lassen sich Sachzusammenhänge oft noch lerngerechter aufbereiten. Als Beispiel soll hier das Teddy-Spiel zum Erlernen objektorientierter Fachkonzepte betrachtet werden. Zusätzlich zu den textuell dargestellten Spielregeln wird ein möglicher Spielablauf mit einem Video aufgezeigt und die Projektvorlage für Greenfoot zum Herunterladen angeboten.

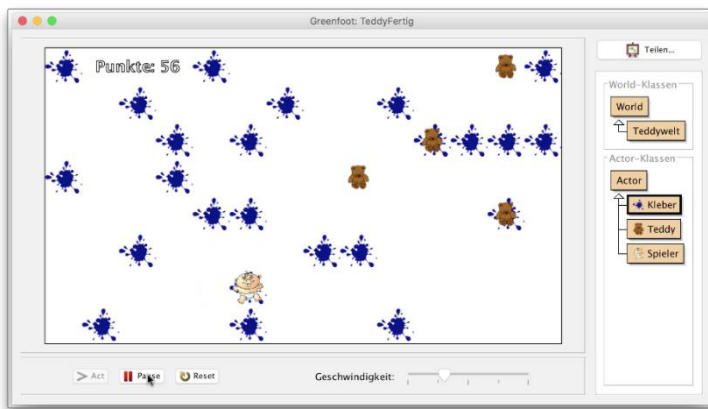


Abb. 2: Das Teddy-Spiel (vgl. <http://www.inf-schule.de/programmierung/oopjava/klassen/aufbau/teddy>)

Aus den Rückmeldungen geht hervor, dass die Integration von multimedialen Inhalten (z.B. Diagramme, Fotos und Videos) von 69,9% der Lernenden als wichtig erachtet wird und die Animation von dynamischen Vorgängen (z.B. Sortierverfahren) von 76,9% der Lernenden. Allerdings wird das Schulbuch inf-schule.de immer noch als textlastig angesehen. Die Bereitschaft, längere Textpassagen zu lesen, ist bei vielen Lernenden nur eingeschränkt vorhanden. Gewünscht wird eine weitere multimediale Aufbereitung. Durch das unbegrenzte Platzangebot in einem elektronischen Schulbuch können weitere Darstellungsformen hinzugefügt werden und vom Lehrenden passend zur Lerngruppe oder auch individuell vom Lernenden passend zu seinem Lernprofil ausgewählt werden. Die multimediale Darstellung ermöglicht es, Lerninhalte noch besser auf die unterschiedlichen Bedürfnisse der Lerner abzustimmen.

4.4 Interaktionsmöglichkeiten

Lernen ist dann besonders effektiv, wenn Lernende sich aktiv-konstruktiv mit den Lerninhalten auseinandersetzen. Zur Erkundung von Informatiksystemen sind insbesondere solche Lernsysteme geeignet, die ein experimentelles Vorgehen ermöglichen, indem einige Parameter des zu erkundenden Systems variiert und die Auswirkungen beobachtet werden. Dadurch lassen sich das Systemverhalten und die zugrundeliegende Systemstruktur vom Lernenden erschließen.

Zum Beispiel können in der Animation zur Erschließung von Dualzahlen im elektronischen Schulbuch die Lernenden interaktiv die binäre Darstellung von Zahlen mit einem Zweizustandssystem erkunden. (vgl. Abb. 3)



Abb. 3: Animation zur Erschließung von Dualzahlen (vgl. http://www.inf-schule.de/information/darstellunginformation/binaerdarstellungzahlen/einstieg_gefaengnisausbruch)

Ein weiteres Beispiel für eine Animation ist auf den Seiten zum Thema Datenbanken zu finden. In einem integrierten SQL-Interpreter, der es ermöglicht, SQL-Befehle zu einer vorgegebenen Datenbank direkt auf inf-schule.de auswerten zu lassen (vgl. z.B. <http://inf-schule-db.cs.uni-kl.de/SQLgbuch1.php>), können die Nutzer ohne Installationsaufwand SQL-Statements testen. Für das Lernen muss also keine lokale Datenbankinfrastruktur aufgebaut werden, diese Infrastruktur wurde direkt in das elektronische Schulbuch „integriert“. Das Schulbuch ist nicht mehr nur Lieferant von Lernmaterial, es wird zur Lernumgebung, die von Lernenden in Teilen aktiv mitgestaltet

werden kann. Die integrierten Werkzeuge stoßen bei 75,6% der befragten Lernenden auf große Zustimmung. Von den Lehrenden wurde insbesondere der nicht vorhandene Installationsaufwand positiv erwähnt. Dies ermöglicht es auch, eine entsprechende Aufgabe als Hausaufgabe bearbeiten zu lassen, weil die Lernenden die Aufgaben mit jedem beliebigen Browser ohne weitere Software bearbeiten können.

4.5 Individualisierung

In der Didaktik gilt der Grundsatz, Lernende möglichst viele Lernwege anzubieten. Gedruckte Schulbücher können dem nur in Ansätzen gerecht werden, weil eine stark begrenzte Ressource in Form von Seiten für das Schulbuch zur Verfügung steht. Solche Beschränkungen gibt es bei einer digitalen Bereitstellung nicht. Das Schulbuch inf-schule.de nutzt diese Möglichkeit, indem es für gleiche Inhalte unterschiedliche didaktische Aufbereitungen anbietet. Lehrende und Lernende haben so Wahlmöglichkeiten und können die für sie geeignetsten Materialien auswählen.

Im Schulbuch inf-schule.de wurden verschiedene Ansätze zur Bereitstellung alternativer Lernwege integriert:

- alternative Programmierparadigmen: control first, data first, objects first, functions first
- alternative didaktische Strukturierungen: systematisch, genetisch, exemplarisch
- alternative kognitive Werkzeuge
- alternative Kontexte
- alternative Ansprüche

Eine genauere Beschreibung dieser Wahlmöglichkeiten wird im elektronischen Schulbuch dargestellt [Be16]. Elektronische Schulbücher ermöglichen es den Autoren viel besser als gedruckte Schulbücher, individualisierte Lernangebote zu machen. Durch erweiterte Ressourcen und mehr Darstellungsmöglichkeiten lassen sich unterschiedliche Lernwege für verschiedene Lerntypen auf differenziertem Anspruchsniveau realisieren. Die unterschiedlichen Darstellungen werden von 73% der Lerner als sehr wichtig erachtet. Als Freitext wurde zum Beispiel zurückgemeldet: „Ferner finde ich es sehr gut, dass die einzelnen [Lernwege] strukturiert aufgebaut sind, um meistens aus einer spezifischen Anfangsproblemstellung durch Analyse und Induktion auf eine allgemeine Aussage zu gelangen.“ Dadurch, dass die allgemeinen Aussagen bei allen Lernwegen identisch sind, können auch in Lerngruppen unterschiedliche Lernwege - passend zum Lernverhalten der Lerner - bearbeitet werden und trotzdem gelangen alle Lerner zum gleichen Ziel. Um auf das unterschiedliche Leistungsniveau der Schüler einzugehen, wurden Aufgaben mit unterschiedlichem Schwierigkeitsgrad integriert. Die verschiedenen Aufgaben wurden zu 71,6% als sehr wichtig bzw. wichtig eingestuft.

4.6 Aktualität der Darstellung

Die digitale Welt ändert sich sehr schnell. Initiativen wie das soziale Netzwerk „schülerVZ“ sind heute kaum noch bekannt und zum Großteil durch andere soziale

Netzwerke ersetzt worden. Im Jahr 2010 hatten die VZ-Netzwerke („schülerVZ“, „studiVZ“ und „meinVZ“) noch über 16 Millionen aktive Nutzer [OI10] und heutzutage ist das soziale Netzwerk den meisten Schülern nicht mehr bekannt. Gedruckte Schulbücher haben es schwer, das Interesse an digitalen Bildungsinhalten mit stets aktuellen Themen zu wecken. Zu groß ist die Gefahr, dass ein Thema sehr schnell aus dem Blickfeld verschwindet.

Digitale Schulbücher bieten dagegen die Möglichkeit, schnell und unkompliziert aktuelle Themen aufzugreifen und ggf. alte durch neuere Themen zu ersetzen. Das ist insbesondere in einem sich dynamisch entwickelnden Fach wie der Informatik, bei dem sich der inhaltliche Kanon noch nicht ganz etabliert hat, von besonderer Bedeutung.

Die Vermittlung von digitaler Bildung erfordert flexible Lernumgebungen. Auch wenn die zu vermittelnden Fachkonzepte stabil bleiben, so müssen die hierzu bereitgestellten Lernumgebungen doch ständig aktualisiert werden. Nur ein digitales Schulbuch kann das leisten, indem ständig Teile des Schulbuchs überarbeitet und ergänzt werden.

Die ständige Weiterentwicklung des elektronischen Schulbuchs inf-schule.de wird von 83,3% der befragten Personen als sehr wichtig erachtet. Die Gefahr der Veränderung, durch die Aktualisierung der Inhalte stellt für die Nutzer kein Problem dar, weil zum Ende eines jeden Schuljahres eine fixierte Version des Schulbuchs archiviert wird. Über einen ergänzenden Parameter beim Aufruf des Schulbuchs kann jede archivierte Version aufgerufen werden und zum Beispiel auch nach drei Jahren zur Prüfungsvorbereitung verwendet werden.

4.7 Kommunikation und Kooperation

Gedruckte Schulbücher werden von kleineren Autorenteamen erstellt. Das hat den Vorteil, dass ein didaktisches Konzept profiliert umgesetzt werden kann. Mehrautorensysteme, wie es zum Beispiel von Wikipedia verwendet wird, verfolgen dagegen die Idee des kollaborierenden Schreibens, bei dem jeder Autor die Möglichkeit hat, am Entwicklungsprozess mitzuwirken. Mit diesem Ansatz wird versucht, das Wissen, Können und die Kreativität von möglichst vielen Autoren zu nutzen. Schwierig ist es dabei, das Gesamtwerk konsistent und einheitlich zu gestalten. Des Weiteren stellt die Qualitätskontrolle, insbesondere bei der Entwicklung eines Schulbuchs, eine große Herausforderung bei einem Mehrautorensystem dar.

Inwieweit sich Mehrautorensysteme für die Schulbuchentwicklung eignen, ist wohl noch nicht hinreichend erprobt. Klar ist, dass ein digitales Schulbuch mit einer digitalen Mehrautoreneninfrastruktur die Schulbuchentwicklung erleichtert.

Das Schulbuchprojekt inf-schule.de hat einen moderaten Weg der Öffnung des Autorenkreises eingeschlagen. Jeder Nutzer kann über ein Rückmeldeformular Hinweise geben, die zu einer partiellen Überarbeitung der Lernmaterialien führen können. Nutzer können auch leicht Kontakt mit dem Autorenteam aufnehmen und Material zur Ergänzung bereitstellen. Nutzer können auf diese Weise direkt zu Mitautoren werden. Die Liste der Autoren ist auf diese Weise auf inzwischen über 50 Personen

angewachsen. Nach wie vor gibt es ein festes Autorenteam, das über die Weiterentwicklung des Schulbuches entscheidet und Qualitätssicherung durch stattfindende Redaktionssitzungen ermöglicht.

Elektronische Schulbücher ermöglichen die Entwicklung für viele Autoren zu öffnen und die Kompetenz im Schulbuch zu vereinen. Durch die vielen Autoren ist der direkte Kontakt zu den Autoren möglich und dies ist für 45,1% der befragten Personen wichtig. Auch die Möglichkeit, einen Beitrag zum elektronischen Schulbuch als Autor beizusteuern, wird von 26,7% der Befragten als wichtig erachtet, wobei die tatsächliche Nutzung bisher deutlich geringer ist.

5 Fazit

Das digitale Schulbuch inf-schule.de kann einen wesentlichen Beitrag zur informatorischen Bildung leisten, indem es wie beschrieben auf unterschiedliche Art die vielfältigen Inhalte der Informatik zur Vermittlung niederschwellig bereitstellt, damit diese mit möglichst niederschweligen Einstiegsvoraussetzungen von Lehrenden gelehrt und von Lernenden erlernt werden können. Denn nur, wenn Lernende in der Schule die Grundlagen zum Verstehen und Gestalten der digitalen Welt erhalten, können diese an der digitalisierten Gesellschaft partizipieren und diese aktiv mitgestalten.

Literaturverzeichnis

- [Be16] Becker, K.: Alternative Didaktisierungen, www.inf-schule.de/infschule/konzeption/didaktischestruktur/didaktisierungen, Stand: 27.01.2017.
- [BB15] Bundesministerium für Bildung und Forschung: Berufsbildungsbericht 2015. www.bmbf.de/pub/Berufsbildungsbericht_2015.pdf, Stand: 29.03.2016.
- [Da16] Dagstuhl-Erklärung: Bildung in der digitalen vernetzten Welt, www.gi.de/fileadmin/redaktion/Themen/dagstuhl-erklaerung-bildung-in-der-digitalen-welt-2016.pdf, Stand: 23.01.2017.
- [Fr16a] Froitzheim, M.; Jonietz, D.; Schneider, O.; Becker, K.; Zimmol, M.: Nutzung eines elektronischen Schulbuchs im Informatikunterricht am Beispiel des Projektes [inf-schule](http://inf-schule.de). In: (Lucke, U.; Schwill, A.; Zender, R): DelfI 2016 – Die 14. E-Learning Tagung Fachtagung Informatik. Bonn: Köllen Druck+Verlag. 2016, S.71-82.
- [Fr16b] Froitzheim, M.; Jonietz, D.; Schneider, O.; Becker, K.; Zimmol, M.: Evaluierung der Akzeptanz des elektronischen Schulbuchs [inf-schule](http://inf-schule.de). In: (Schuhen, M; Froitzheim, M.; Schuhen, K.): Das elektronische Schulbuch 2016. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik, Münster 2016, S. 83-98.

- [HP10] Humbert, L.; Panske, J.: Informatik und Gender nimmt die Forschungsergebnisse ernst!, <http://bscw.ham.nw.schule.de/pub/bscw.cgi/d2009202/HumbertPanske2010.pdf>, Stand: 29.03.2016.
- [Pr12a] Prediger, S.; Link, M.: Fachdidaktische Entwicklungsforschung – ein entwicklungsprozessfokussierendes Forschungsprogramm mit Verschränkung fachdidaktischer Arbeitsbereich. In: (Bayrhuber, H.; Harms, U.; Muszynski, B.; Ralle, B.; Rothgangel, M.; Schön, L.-H.; Vollmer, H.; Weigand, H.-G.): Formate Fachdidaktischer Forschung. Fachdidaktische Forschungen, Bd. 2, Münster, S. 29-46.
- [Pr12b] Prediger, S.; Link, M.; Hinz, R.; Hußmann, S.; Thiele, J.; Ralle, B.: Lehr-Lernprozesse initiieren und erforschen – Fachdidaktische Entwicklungsforschung im Dortmunder Modell. In: MNU 65 (8), S. 452-457.
- [OI10] Online ich: Top 100 Soziale Netzwerke in Deutschland, <http://online-ich.de/20100125/top-100-soziale-netzwerke-deutschland/>, Stand: 27.01.2017.
- [Sa10] Sandfuchs, U. (2010). Schulbücher und Unterrichtsqualität - historische und aktuelle Reflexionen. In: (Fuchs, E.; Kahlert, J. ; Sandfuchs, U. (Hg.)): Schulbuch konkret. Bad Heilbrunn: Julius Klinkhardt, 11-24.
- [Sa12] Sanguo, C.; Xuehai, M.; Schenglin, L. (2012). The Pace of Ebook Development in China. IN: LOGOS: The Journal of the World Book Community, 23(2), 14-20.
- [ScFr14] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [ScFr15] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch 2015. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [ScFr16] Schuhen, M.; Froitzheim, M. (Hrsg.): Das Elektronische Schulbuch 2016. Fachdidaktische Anforderungen und Ideen treffen auf Lösungsvorschläge der Informatik. Münster: LIT Verlag.
- [JIM15] Behrens, P.; Rathgeb, T.; Feierabend, S.; Plankenhorn, T: JIM 2015 - Jugend, Information, (Multi-) Media. Basisstudie zum Medienumgang 12 - bis 19-Jähriger in Deutschland. Stuttgart: Medienpädagogischer Forschungsverbund Südwest. www.mpfs.de/fileadmin/files/Studien/JIM/2015/JIM_Studie_2015.pdf, Stand: 29.03.2016.
- [Wi03] Wiater, W. (2003): Schulbuchforschung in Europa. Bestandsaufnahme und Zukunftsperspektive. Bad Heilbrunn: Klinkhardt.

Autorenverzeichnis

A

Akao, Kensuke, 405

B

Barkmin, Mike, 407
Becker, Klaus, 345, 359
Becker, Klaus-Peter, 443
Benedikt, Gregor Alexander, 363
Bergner, Nadine, 53, 63, 237
Bescherer, Christine, 409
Best, Alexander, 83
Borsch, Johanna, 211
Braun, Friederike, 119
Brinda, Torsten, 119, 321, 393, 407,
411, 425
Brinkmeier, Michael, 311, 349

C

Class, Christina B., 349
Czechowski, Robert, 431

D

Dengel, Andreas, 87
Diekmann, Leonard, 181
Dietrich, Leonore, 191, 373
Dittert, Nadine, 287, 291
Duwe, Wibke, 413

E

Engbring, Dieter, 227

F

Fest, Andreas, 409
Fischer, Johannes, 247
Förster, Klaus-Tycho, 171
Fricke, Martin, 429

Froitzheim, Manuel, 345, 359, 443
Funke, Alexandra, 327
Futschek, Gerald, 427

G

Gallenbacher, Jens, 19, 267
Gärtig-Daug, Anja, 419
Geldreich, Katharina, 327
Goecke, Lennart, 417
Gold-Veerkamp, Carolin, 393
Grillenberger, Andreas, 157
Grimm, Rüdiger, 167
Günther, Christine, 415

H

Haji, Lina, 363
Hancl, Mirek, 421
Haselmeier, Kathrin, 429
Heuer, Ute, 87
Heun, Dominik, 201
Hielscher, Michael, 97
Hildebrandt, Claudia, 137
Hoffmann, Sabrina, 73
Honegger, Beat Döbeli, 97
Hubwieser, Peter, 327
Hug, Alexander, 167
Humbert, Ludger, 429

J

Jäckel, Stefanie, 423
Jatzlau, Sven, 383
Jochum, Heiko, 345, 359, 443
Jonietz, Daniel, 345
Jördens, Tobias, 267
Jückstock, Petra, 335

K

Kastl, Petra, 301
Katterfeldt, Eva-Sophie, 287, 291
Knaus, Thomas, 31
Knobelsdorf, Maria, 181
Köster, Hilde, 53
Kramer, Matthias, 407, 425
Krisch, Oliver, 301

L

Leonhardt, Thiemo, 237
Levens, René, 363
Lindner, Marlene, 277

M

Magenheim, Johannes, 53
Meyer, Hilbert, 29
Micheuz, Peter, 427
Mühling, Andreas, 123
Müller, Dorothee, 127, 429
Müller, Kathrin, 53, 109
Müller, Olaf, 363

N

Noller, Stephan, 49

O

Otto, Jonathan, 181

P

Pancratz, Nils, 393
Pasternak, Arno, 247
Pech, Detlef, 417
Petrut, Sandra-Jasmin, 63
Pinkwart, Niels, 277, 393, 417
Pisarenko, Stanislav, 311
Pohl, Wolfgang, 431
Prädel, Lars, 221
Przybylla, Mareen, 257

R

Ribe, Liz, 349

Rohde, Thomas, 363
Romeike, Ralf, 53, 157, 257, 301, 383
Romero, Esther Alzate, 191, 373
Roßner, Marc, 433
Rücker, Michael T., 393

S

Saathoff, Arne, 435
Samimi, Vahid, 425
Schmid, Ute, 419
Schneider, Yannick, 123
Schreiber, Gerlinde, 221
Schroeder, Ulrik, 53, 63, 237
Schulte, Carsten, 53, 109
Schulz, Sandra, 277
Schwinem, Stefan, 321
Siebrecht, Daniel, 429, 437
Steinbach, Nico, 147
Steinhäuser, Anke, 439
Stierle, Maarit, 441
Stiller, Jurik, 417
Strauss, Kolja, 363

T

Terjung, Thorsten, 411
Thiele, Otto, 335
Thomas, Marco, 211
Tobinski, David, 321, 407

W

Weitz, Katharina, 419
Welke, Michel, 181
Wendlandt, Katharina, 73
Wendlandt, Matthias, 73
Wieczorek, Barbara, 349

Z

Zimmol, Martin, 345, 359, 443
Zitzler, Eckart, 147

GI-Edition Lecture Notes in Informatics

- P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.
- P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.
- P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühlhng, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensor-gestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelpath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.
- P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.
- P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.
- P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.
- P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
- P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.
- P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.
- P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.
- P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.
- P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
- P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.
- P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.
- P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.
- P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.
- P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).
- P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.
- P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.
- P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit
- P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen
- P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003
- P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web
- P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin
- P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Erfahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

- P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003
- P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft
- P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenber (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
- P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
- P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003
- P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
- P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004
- P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing
- P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
- P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning, E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven
- P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
- P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
- P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
- P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government
- P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
- P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
- P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Rannenber, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung
- P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen
- P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit
- P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and its Applications

- P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolfrid Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web
- P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik
- P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)
- P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)
- P-69 Robert Hirschfeld, Ryszard Kowalczyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005
- P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)
- P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005
- P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment
- P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): "Heute schon das Morgen sehen"
- P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology
- P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture
- P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz
- P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006
- P-78 K.-O. Wenkel, P. Wagner, M. Morgens-tern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel
- P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006
- P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce
- P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS'06
- P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006
- P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics
- P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications
- P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems
- P-86 Robert Krimmer (Ed.): Electronic Voting 2006
- P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik
- P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006
- P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur
- P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006
- P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006
- P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1
- P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2
- P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen
- P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies
- P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

- P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices
- P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik
- P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik
- P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten
- P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments
- P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)
- P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme
- P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007
- P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops
- P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications
- P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures
- P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1
- P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2
- P-111 Christian Eibl, Johannes Magenheimer, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik
- P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis
- P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)
- P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics
- P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walthert (Eds.) German conference on bioinformatics GCB 2007
- P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007
- P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and management The 4th International Conference on Grid Service Engineering and Management GSEM 2007
- P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007
- P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications
- P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges
- P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik
- P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

- P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)
- P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Special Interest Groups PARS and PARVA
- P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT:
Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung
- P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)
- P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008
- P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany
- P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems
- P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung
- P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik und E-Voting, CC
- P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik
- P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik Band 1
- P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik Band 2
- P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik –
Aktuelle Forschungsergebnisse
- P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008
- P-137 Arslan Brömme, Christoph Busch, Detlef Hühlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures
- P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008
- P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik
- P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics
- P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management
- P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung
- P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

- P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)
- P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management
- P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)
- P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009
- P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I²CS 2009
- P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum
Kommunikationstechnologien
Beiträge der Fachtagung
- P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband
- P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software
- P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009
- P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik
- P-154 Stefan Fischer, Erik Maehle, Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben
- P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures
- P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«
- P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009
- P-158 W. Claudepein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture
Reloaded – Informationsgestützte Landwirtschaft
- P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010
- P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 –
Workshopband
(inkl. Doktorandensymposium)
- P-161 Gregor Engels, Dimitris Karagiannis, Heinrich C. Mayr (Hrsg.)
Modellierung 2010
- P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung
Verwaltungsinformatik (FTVI) und
Fachtagung Rechtsinformatik (FTRI) 2010
- P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und
Dienste zur Unterstützung von mobiler
Kollaboration
- P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

- P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I²CS) – Jubilee Edition 2010 –
- P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien Beiträge der Fachtagung
- P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC
- P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik
- P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.
- P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit
- P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management
- P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA
(Entwicklungsmethoden für Informationssysteme und deren Anwendung)
- P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010
- P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGktive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“
- P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1
- P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2
- P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC
- P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010
- P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM'2010
Second International Workshop on Formal Methods and Agile Methods
- P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)
- P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft
- P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action
- P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

- P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)
- P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I²CS)
- P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikationstechnologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn
- P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden
- P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf INFOS 2011
14. GI-Fachtagung Informatik und Schule
- P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)
- P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group
- P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities
- P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011
- P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landwirtschaft Fokus Forstwirtschaft
- P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
- P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group
- P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012
- P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband
- P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops
- P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012
- P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012: Mobile und Ubiquitäre Informationssysteme
- P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung
- P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I²CS 2012)
- P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC
- P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung
- P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

- P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.)
INFORMATIK 2012
- P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.)
Nachhaltiges Software Management
- P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.)
Automotive – Safety & Security 2012
Sicherheit und Zuverlässigkeit für automobile Informationstechnik
- P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.)
Massendatenmanagement in der Agrar- und Ernährungswirtschaft
Erhebung - Verarbeitung - Nutzung
Referate der 33. GIL-Jahrestagung
20. – 21. Februar 2013, Potsdam
- P-212 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2013
Proceedings of the 12th International Conference of the Biometrics Special Interest Group
04.–06. September 2013
Darmstadt, Germany
- P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.)
Software Engineering 2013
Fachtagung des GI-Fachbereichs Softwaretechnik
- P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mitschang, Theo Härder, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013
13. – 15. März 2013, Magdeburg
- P-215 Stefan Wagner, Horst Lichter (Hrsg.)
Software Engineering 2013
Workshopband
(inkl. Doktorandensymposium)
26. Februar – 1. März 2013, Aachen
- P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband
11. – 12. März 2013, Magdeburg
- P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
6. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung
03.–04. Juni 2013, Erlangen
- P-218 Andreas Breiter, Christoph Rensing (Hrsg.)
DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
8. – 11. September 2013, Bremen
- P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.)
Informatik erweitert Horizonte
INFOS 2013
15. GI-Fachtagung Informatik und Schule
26. – 28. September 2013
- P-220 Matthias Horbach (Hrsg.)
INFORMATIK 2013
Informatik angepasst an Mensch, Organisation und Umwelt
16. – 20. September 2013, Koblenz
- P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimos Tambouris (Eds.)
Electronic Government and Electronic Participation
Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013
16. – 19. September 2013, Koblenz
- P-222 Reinhard Jung, Manfred Reichert (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2013)
St. Gallen, Switzerland
September 5. – 6. 2013
- P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2013
10. – 11. September 2013
Kloster Banz, Germany
- P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.)
Vorgehensmodelle 2013
Vorgehensmodelle – Anspruch und Wirklichkeit
20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V.
Lörrach, 2013
- P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.)
Modellierung 2014
19. – 21. März 2014, Wien
- P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.)
IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement
Referate der 34. GIL-Jahrestagung
24. – 25. Februar 2014, Bonn

- P-227 Wilhelm Hasselbring,
Nils Christian Ehmke (Hrsg.)
Software Engineering 2014
Fachtagung des GI-Fachbereichs
Softwaretechnik
25. – 28. Februar 2014
Kiel, Deutschland
- P-228 Stefan Katzenbeisser, Volkmar Lotz,
Edgar Weippl (Hrsg.)
Sicherheit 2014
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 7. Jahrestagung des
Fachbereichs Sicherheit der
Gesellschaft für Informatik e.V. (GI)
19. – 21. März 2014, Wien
- P-229 Dagmar Lück-Schneider, Thomas
Gordon, Siegfried Kaiser, Jörn von
Lucke, Erich Schweighofer, Maria
A. Wimmer, Martin G. Löhle (Hrsg.)
Gemeinsam Electronic Government
ziel(gruppen)gerecht gestalten und
organisieren
Gemeinsame Fachtagung
Verwaltungsinformatik (FTVI) und
Fachtagung Rechtsinformatik (FTRI)
2014, 20.-21. März 2014 in Berlin
- P-230 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2014
Proceedings of the 13th International
Conference of the Biometrics Special
Interest Group
10. – 12. September 2014 in
Darmstadt, Germany
- P-231 Paul Müller, Bernhard Neumair,
Helmut Reiser, Gabi Dreo Rodosek
(Hrsg.)
7. DFN-Forum
Kommunikationstechnologien
16. – 17. Juni 2014
Fulda
- P-232 E. Plödereder, L. Grunske, E. Schneider,
D. Ull (Hrsg.)
INFORMATIK 2014
Big Data – Komplexität meistern
22. – 26. September 2014
Stuttgart
- P-233 Stephan Trahasch, Rolf Plötzner, Gerhard
Schneider, Claudia Gayer, Daniel Sassiati,
Nicole Wöhrle (Hrsg.)
DeLFI 2014 – Die 12. e-Learning
Fachtagung Informatik
der Gesellschaft für Informatik e.V.
15. – 17. September 2014
Freiburg
- P-234 Fernand Feltz, Bela Mutschler, Benoît
Ottjacques (Eds.)
Enterprise Modelling and Information
Systems Architectures
(EMISA 2014)
Luxembourg, September 25-26, 2014
- P-235 Robert Giegerich,
Ralf Hofestädt,
Tim W. Nattkemper (Eds.)
German Conference on
Bioinformatics 2014
September 28 – October 1
Bielefeld, Germany
- P-236 Martin Engstler, Eckhart Hanser,
Martin Mikusz, Georg Herzwurm (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2014
Soziale Aspekte und Standardisierung
Gemeinsame Tagung der Fachgruppen
Projektmanagement (WI-PM) und
Vorgehensmodelle (WI-VM) im
Fachgebiet Wirtschaftsinformatik der
Gesellschaft für Informatik e.V., Stuttgart
2014
- P-237 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2014
4.–6. November 2014
Stuttgart, Germany
- P-238 Arno Ruckelshausen, Hans-Peter
Schwarz, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Referate der 35. GIL-Jahrestagung
23. – 24. Februar 2015, Geisenheim
- P-239 Uwe Aßmann, Birgit Demuth, Thorsten
Spitta, Georg Püschel, Ronny Kaiser
(Hrsg.)
Software Engineering & Management
2015
17.-20. März 2015, Dresden
- P-240 Herbert Klenk, Hubert B. Keller, Erhard
Plödereder, Peter Dencker (Hrsg.)
Automotive – Safety & Security 2015
Sicherheit und Zuverlässigkeit für
automobile Informationstechnik
21.–22. April 2015, Stuttgart
- P-241 Thomas Seidl, Norbert Ritter,
Harald Schöning, Kai-Uwe Sattler,
Theo Härder, Steffen Friedrich,
Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2015)
04. – 06. März 2015, Hamburg

- P-242 Norbert Ritter, Andreas Henrich, Wolfgang Lehner, Andreas Thor, Steffen Friedrich, Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2015) – Workshopband
02. – 03. März 2015, Hamburg
- P-243 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
8. DFN-Forum
Kommunikationstechnologien
06.–09. Juni 2015, Lübeck
- P-244 Alfred Zimmermann, Alexander Rossmann (Eds.)
Digital Enterprise Computing (DEC 2015)
Böblingen, Germany June 25-26, 2015
- P-245 Arslan Brömme, Christoph Busch, Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2015
Proceedings of the 14th International Conference of the Biometrics Special Interest Group
09.–11. September 2015
Darmstadt, Germany
- P-246 Douglas W. Cunningham, Petra Hofstedt, Klaus Meer, Ingo Schmitt (Hrsg.)
INFORMATIK 2015
28.9.-2.10. 2015, Cottbus
- P-247 Hans Pongratz, Reinhard Keil (Hrsg.)
DeLFI 2015 – Die 13. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
1.–4. September 2015
München
- P-248 Jens Kolb, Henrik Leopold, Jan Mendling (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 6th Int. Workshop on Enterprise Modelling and Information Systems Architectures, Innsbruck, Austria
September 3-4, 2015
- P-249 Jens Gallenbacher (Hrsg.)
Informatik
allgemeinbildend begreifen
INFOS 2015 16. GI-Fachtagung
Informatik und Schule
20.–23. September 2015
- P-250 Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Martin Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und Vorgehensmodelle 2015
Hybride Projektstrukturen erfolgreich umsetzen
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V., Elmshorn 2015
- P-251 Detlef Hühnlein, Heiko Roßnagel, Raik Kuhlisch, Jan Ziesing (Eds.)
Open Identity Summit 2015
10.–11. November 2015
Berlin, Germany
- P-252 Jens Knoop, Uwe Zdun (Hrsg.)
Software Engineering 2016
Fachtagung des GI-Fachbereichs Softwaretechnik
23.–26. Februar 2016, Wien
- P-253 A. Ruckelshausen, A. Meyer-Aurich, T. Rath, G. Recke, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Fokus: Intelligente Systeme – Stand der Technik und neue Möglichkeiten
Referate der 36. GIL-Jahrestagung
22.-23. Februar 2016, Osnabrück
- P-254 Andreas Oberweis, Ralf Reussner (Hrsg.)
Modellierung 2016
2.–4. März 2016, Karlsruhe
- P-255 Stefanie Betz, Ulrich Reimer (Hrsg.)
Modellierung 2016 Workshopband
2.–4. März 2016, Karlsruhe
- P-256 Michael Meier, Delphine Reinhardt, Steffen Wendzel (Hrsg.)
Sicherheit 2016
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
5.–7. April 2016, Bonn
- P-257 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
9. DFN-Forum
Kommunikationstechnologien
31. Mai – 01. Juni 2016, Rostock

- P-258 Dieter Hertweck, Christian Decker (Eds.)
Digital Enterprise Computing (DEC 2016)
14.–15. Juni 2016, Böblingen
- P-259 Heinrich C. Mayr, Martin Pinzger (Hrsg.)
INFORMATIK 2016
26.–30. September 2016, Klagenfurt
- P-260 Arslan Brömme, Christoph Busch,
Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2016
Proceedings of the 15th International
Conference of the Biometrics Special
Interest Group
21.–23. September 2016, Darmstadt
- P-261 Detlef Rätz, Michael Breidung, Dagmar
Lück-Schneider, Siegfried Kaiser, Erich
Schweighofer (Hrsg.)
Digitale Transformation: Methoden,
Kompetenzen und Technologien für die
Verwaltung
Gemeinsame Fachtagung
Verwaltungsinformatik (FTVI) und
Fachtagung Rechtsinformatik (FTRI) 2016
22.–23. September 2016, Dresden
- P-262 Ulrike Lucke, Andreas Schwill,
Raphael Zender (Hrsg.)
DeLFI 2016 – Die 14. E-Learning
Fachtagung Informatik
der Gesellschaft für Informatik e.V. (GI)
11.–14. September 2016, Potsdam
- P-263 Martin Engstler, Masud Fazal-Baqaie,
Eckhart Hanser, Oliver Linssen, Martin
Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und
Vorgehensmodelle 2016
Arbeiten in hybriden Projekten: Das
Sowohl-als-auch von Stabilität und
Dynamik
Gemeinsame Tagung der Fachgruppen
Projektmanagement (WI-PM) und
Vorgehensmodelle (WI-VM) im
Fachgebiet Wirtschaftsinformatik
der Gesellschaft für Informatik e.V.,
Paderborn 2016
- P-264 Detlef Hühnlein, Heiko Roßnagel,
Christian H. Schunck, Maurizio Talamo
(Eds.)
Open Identity Summit 2016
der Gesellschaft für Informatik e.V. (GI)
13.–14. October 2016, Rome, Italy
- P-265 Bernhard Mitschang, Daniela
Nicklas, Frank Leymann, Harald
Schöning, Melanie Herschel, Jens
Teubner, Theo Härder, Oliver Kopp,
Matthias Wieland (Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2017)
6.–10. März 2017, Stuttgart
- P-266 Bernhard Mitschang, Norbert Ritter,
Holger Schwarz, Meike Klettke, Andreas
Thor, Oliver Kopp, Matthias Wieland
(Hrsg.)
Datenbanksysteme für Business,
Technologie und Web (BTW 2017)
Workshopband
6.–7. März 2017, Stuttgart
- P-267 Jan Jürjens, Kurt Schneider (Hrsg.)
Software Engineering 2017
21.–24. Februar 2017, Hannover
- P-268 A. Ruckelshausen, A. Meyer-Aurich,
W. Lentz, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und
Ernährungswirtschaft
Fokus: Digitale Transformation –
Wege in eine zukunftsfähige
Landwirtschaft
Referate der 37. GIL-Jahrestagung
06.–07. März 2017, Dresden
- P-269 Peter Dencker, Herbert Klenk, Hubert
Keller, Erhard Plödereder (Hrsg.)
Automotive – Safety & Security 2017
30.–31. Mai 2017, Stuttgart
- P-271 Paul Müller, Bernhard Neumair, Helmut
Reiser, Gabi Dreö Rodosek (Hrsg.)
10. DFN-Forum Kommunikations-
technologien
30. – 31. Mai 2017, Berlin
- P-272 Alexander Rossmann, Alfred
Zimmermann (eds.)
Digital Enterprise Computing
(DEC 2017)
11.–12. Juli 2017, Böblingen
- P-273 Christoph Igel, Carsten Ullrich,
Martin Wessner (Hrsg.)
BILDUNGSRÄUME
DeLFI 2017
Die 15. e-Learning Fachtagung Informatik
der Gesellschaft für Informatik e.V. (GI)
5. bis 8. September 2017, Chemnitz

- P-274 Ira Diethelm (Hrsg.)
Informatische Bildung zum Verstehen
und Gestalten der digitalen Welt
13.–15. September 2017
Oldenburg
- P-277 Lothar Fritsch, Heiko Roßnagel,
Detlef Hühnlein (Hrsg.)
Open Identity Summit 2017
5.– 6. October 2017
Karlstad, Sweden

The titles can be purchased at:
Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de

Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into

- seminars
- proceedings
- dissertations
- thematics

current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: <http://www.gi.de/service/publikationen/lni/>

ISSN 1617-5468

ISBN 978-3-88579-668-8

“Informatische Bildung zum Verstehen und Gestalten der digitalen Welt – INFOS 2017” ist die 17. Konferenz der Reihe Informatik und Schule, die ihren Fokus auf das breite Spektrum der informatischen Bildung allgemein und der Informatik als Schulfach im Speziellen legt. Dieser Band steht im Kontext der bundesweiten Diskussionen zur Bildung in der digitalen Welt und zeichnet sich daher durch eine große Spannweite aus – sowohl in der Gruppe der Lerner beginnend mit dem Kindergarten als auch in den behandelten Themengebieten. Der Tradition folgend legt auch diese viel Wert drauf, sowohl Theorie und Forschung als auch Praxisbeiträgen angemessenen Raum zu geben.