

Gesellschaft für Informatik (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 the fields of

- Seminar
- Proceedings
- Dissertations
- Thematics

current topics are dealt with from the fields of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure the high level of the contributions.

The volumes are published in German or English.

Information: <http://www.gi-ev.de/service/publikationen/lni/>

ISSN 1617-5468

ISBN 978-3-88579-250-5

»INFOS 2009« is the 13th event in a conference series organized by the GI special interest group IBS focusing on the informatics education in schools, which includes the discussion of teaching concepts, educational uses of information and communication technologies and computer based media.

The first event in this conference series took place 25 years ago at the Free University of Berlin (Freie Universität Berlin). The present volume contains contributions accepted for the 2009 conference at the same place according to this anniversary. (in German)



B. Koerber (Hrsg.): Zukunft braucht Herkunft – 25 Jahre INFOS

156



GI-Edition

Lecture Notes in Informatics

Bernhard Koerber (Hrsg.)

Zukunft braucht Herkunft

25 Jahre »INFOS – Informatik und Schule«

INFOS 2009

**13. GI-Fachtagung Informatik und Schule
21. – 24. September 2009 in Berlin**

Proceedings



Bernhard Koerber (Hrsg.)

**Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«**

**INFOS 2009
13. GI-Fachtagung »Informatik und Schule«**

**21. bis 24. September 2009
an der Freien Universität Berlin**

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) – Proceedings
Series of the Gesellschaft für Informatik (GI)

Volume P-156

ISBN 978-3-88579-250-5

ISSN 1617-5468

Volume Editor

Bernhard Koerber

Freie Universität Berlin

Fachbereich Erziehungswissenschaft und Psychologie

Gemeinsame Einrichtung Datenverarbeitung und informatische Bildung

Habelschwerdter Allee 45

14195 Berlin, Germany

E-Mail: bernhard.koerber@fu-berlin.de

Series Editorial Board

Heinrich C. Mayr, Universität Klagenfurt, Austria (Chairman, mayr@ifit.uni-klu.ac.at)

Hinrich Bonin, Leuphana-Universität Lüneburg, Germany

Dieter Fellner, Technische Universität Darmstadt, Germany

Ulrich Flegel, SAP Research, Germany

Ulrich Frank, Universität Duisburg-Essen, Germany

Johann-Christoph Freytag, Humboldt-Universität Berlin, Germany

Thomas Roth-Berghofer, DFKI

Michael Goedicke, Universität Duisburg-Essen

Ralf Hofestädt, Universität Bielefeld

Michael Koch, Universität der Bundeswehr, München, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Ernst W. Mayr, Technische Universität München, Germany

Sigrid Schubert, Universität Siegen, Germany

Martin Warnke, Leuphana-Universität Lüneburg, Germany

Dissertations

Dorothea Wagner, Universität Karlsruhe, Germany

Seminars

Reinhard Wilhelm, Universität des Saarlandes, Germany

Thematics

Andreas Oberweis, Universität Karlsruhe (TH)

© Gesellschaft für Informatik, Bonn 2009

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

Vor 25 Jahren – im Oktober 1984 – wurde an der Freien Universität Berlin die erste Tagung dieser Reihe zur Informatik in der Schule durchgeführt. »Die Breite der angebotenen Referate auf der nunmehr in Berlin stattfindenden Fachtagung macht deutlich«, so schrieb der damalige Tagungsleiter Wolfgang Arlt in seinem Vorwort zum Tagungsband¹ [AH84, S. IV], »daß die Integration der Informatik in Schule und Ausbildung in Gang gekommen ist – allerdings auch jetzt noch mit völlig unzureichenden Mitteln und Methoden. Trotz allem besteht die Chance, mit solchen Fachtagungen insbesondere Lehrern und Ausbildern einen Überblick über Erfahrungen, Anregungen und Hilfen zur Informatik und Datenverarbeitung in Schule und Ausbildung zu geben.«

An diesem Grundsatz hat sich nichts geändert, was wohl den Erfolg und die mittlerweile 25-jährige Tradition dieser Tagungsreihe erklärt. So stellte zur 10. Fachtagung im Jahr 2003 Peter Hubwieser als Tagungsleiter ebenfalls im Vorwort zum Tagungsband² anlässlich dieses Jubiläums fest [Hu03, S. 3]: »Die ›INFOS‹ des Fachausschusses ›Informatische Bildung an Schulen‹ (IBS, früher 7.3) ist zum festen Begriff in der Landschaft der informatischen Bildung Deutschlands geworden [...].«

Sicherlich lassen sich erste Ansätze zur Integration informatischer Inhalte in den Unterricht allgemeinbildender und beruflicher Schulen bereits bis in die Mitte der Sechzigerjahre des vorigen Jahrhunderts zurück verfolgen. Doch erst als die Ständige Konferenz der Kultusminister der Länder der Bundesrepublik Deutschland am 7. Juli 1972 in ihrer »Vereinbarung zur Neugestaltung der gymnasialen Oberstufe in der Sekundarstufe II« unter anderem beschloss, Informatik als Grundkurs zuzulassen, gab es den nötigen Impuls für Lehrerinnen und Lehrer, für Wissenschaftlerinnen und Wissenschaftler und sogar für Bildungspolitikern und -politiker, sich intensiver mit diesem Fach auseinanderzusetzen. Förderprogramme und Modellversuche wurden angestoßen, Ideen konnten entwickelt, Erfahrungen ausgewertet und systematisiert werden. Der notwendige intensive Austausch solcher Ideen und Erfahrungen kann aber letztlich erst – damals wie heute – vor allem durch Tagungen, das heißt aufgrund der persönlichen Begegnung der in diesem Bereich Aktiven in Gang gesetzt werden.

Bereits im 1971 beschlossenen Zweiten Datenverarbeitungsprogramm der Bundesregierung war vorgesehen, Fördermittel für den Einsatz der »Datenverarbeitung im Bildungsbereich« – wie es zu jener Zeit hieß – bereitzustellen. Zwar war das Hauptziel dieses Teils des Programms, die Informatik an den Hochschulen zu fördern, doch ebenso war den Beteiligten klar, dass bereits vor dem Studium in den Schulen entsprechende Anre-

¹ [AH84] Arlt, W.; Haefner, K. (Hrsg.): Informatik als Herausforderung an Schule und Ausbildung – GI-Fachtagung, Berlin, 8.–10. Oktober 1984. Reihe »Informatik-Fachberichte«, Band 90. Springer-Verlag, Berlin u. a., 1984.

² [Hu03] Hubwieser, P. (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003 – 10. GI-Fachtagung Informatik und Schule, 17.–19. September 2003 in Garching bei München. Reihe GI-Edition »Lecture Notes in Informatics«, Band P-32. Köllen Verlag, Bonn, 2003.

gungen gegeben werden müssen, sich mit dem Fach und seinen Anwendungen auseinanderzusetzen. Eine in Paderborn am damaligen Forschungs- und Entwicklungszentrum für objektivierte Lern- und Lehrverfahren (FEoLL) vom Bund angesiedelte Projektgruppe – der »Projekträger« – diente der Steuerung der geförderten Projekte. Von den Projektdurchführenden wurde ein »Arbeitskreis Computerunterstützter Unterricht« (ACU) gegründet, über den alle halbe Jahre eine Tagung bei einer der geförderten Institutionen organisiert und dokumentiert wurde. Einen solchen intensiven Austausch auf dieser breiten Basis hat es im Übrigen nie wieder gegeben. Die Mehrzahl dieser Projekte war zunächst auf den Einsatz von Computern als Unterrichtsmedium ausgerichtet, aber bald wurde deutlich, dass sowohl die Lernenden als auch die Lehrenden mehr über Computer erfahren wollten, als dies in den bisherigen Projektansätzen vorgesehen war. Bereits 1974 wurde dies auf einer Tagung in Hamburg mit dem Titel »Rechner-Gestützter Unterricht« deutlich. Zwar konstatierte der Tagungsleiter Klaus Brunnstein im Vorwort zum Tagungsband³ [BHH74, S. II]: »Auch das Thema ›Schulfach Informatik‹ kann leider nur am Rande der Tagung in einer Podiumsdiskussion ›Computer und Schule‹ behandelt werden, obwohl dieses Fach mit dem Einsatz des Rechners als Lehr- und Lernmedium eng zusammenhängt, weil nämlich Strategien wie ›Interaktives Programmieren‹ und ›Problemlösen‹ im RGU oft ohne Informatikkenntnisse bei Lehrern und Schülern nicht eingesetzt werden können.« Doch letztlich setzte sich das Schulfach Informatik und kaum der computerunterstützte Unterricht in den Schulen durch, und der ACU nannte sich dann bald in »Arbeitskreis Computer im Unterricht« um.

Blieb jedoch in den Siebzigerjahren des vorigen Jahrhunderts die Informatik als Schulfach noch eine Besonderheit, so sollte sich dies Anfang der Achtzigerjahre ändern. Eine der Initialzündungen dazu war unter anderem eine Expertentagung zum Thema »Neue Technologien und Schule«, die im Oktober 1983 von der Evangelischen Akademie Loccum durchgeführt und dokumentiert⁴ wurde. Jeder Lehrende sei aufgefordert, »diesen neuen Themenbereich für sich und seine Unterrichtsfächer fachlich und didaktisch zu erschließen und weiterzuentwickeln«, wurde dort für notwendig erklärt, und jeder Lehrende, die oder der solche Initiativen ergreife, handle im Sinne der Bemühungen und solle daher »nach Kräften unterstützt und ermutigt werden«, hieß es weiter [Loc83, S. 204-205].

Auf einer Fachtagung⁵ in Wiesbaden im März 1984 äußerte schließlich die damalige Bundesministerin für Bildung und Wissenschaft, Dorothee Wilms [PHD84, S. 18], dass »wegen der breiten Diffusion der Informationstechniken in alle Lebensbereiche [...] informationstechnisches Wissen kein Privileg von Spezialisten bleiben [kann]. Es ist

³ [BHH74] Brunnstein, K.; Haefner, K.; Händler, W. (Hrsg.): Rechner-Gestützter Unterricht – RGU '74 – Fachtagung, Hamburg, 12.–14. August 1974. Reihe »Lecture Notes in Computer Science«, Band 17. Springer-Verlag, Berlin u. a., 1974.

⁴ [Loc83] Evangelische Akademie Loccum; Niedersächsisches Kultusministerium (Hrsg.): Neue Technologien und Schule. Reihe »Loccumer Protokolle«, 23/1983. Rehburg-Loccum, 1984.

⁵ [PHD84] Peschke, R.; Hullen, G.; Diemer, W. R. (Hrsg.): Anforderungen an neue Lernerhalte. Band 1: Ergebnisse der Fachtagung »Mikroelektronik und Schule III«, Wiesbaden, 27. und 28. März 1984. Band 2: Sachstandsberichte zum Informatik-Unterricht in der Bundesrepublik Deutschland. Reihe »Materialien zur Schulentwicklung«, Heft 4 und Heft 5. HIBS – Hessisches Institut für Bildungsplanung und Schulentwicklung, Wiesbaden, 1984.

vielmehr zu einem notwendigen Bestandteil einer Allgemeinbildung geworden, über die jeder verfügen muß, um in Alltag und Arbeitswelt kompetent und verantwortungsbewußt handeln zu können. So wird auch erst für eine ausreichende Vorbereitung auf die Teilnahme an dem in einer freien Gesellschaft und in der Demokratie unerläßlichen Informations- und Willensbildungsprozeß angesichts der zukünftigen Fülle der Informationsmöglichkeiten Sorge getragen. Dies wird nur möglich sein, wenn die Vermittlung dieses Grundwissens als allgemeinbildende Aufgabe verstanden wird, zu deren wesentlichen Merkmalen es gehört, daß die fachlichen Inhalte in größere Zusammenhänge und Wertorientierungen eingebettet sind.«

Schließlich veröffentlichte 1984 die Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung (BLK) ein »Rahmenkonzept für die informationstechnische Bildung in Schule und Ausbildung«, das von ihr 1987 durch ein »Gesamtkonzept für die informationstechnische Bildung« vervollständigt und abgelöst wurde⁶. Kernpunkt der Forderungen des Gesamtkonzepts der BLK war eine Grundbildung für alle Schülerinnen und Schüler und darüber hinaus eine »vertiefende informationstechnische Bildung«, d. h. eine freiwillige Fortführung der Grundbildung in einem Wahlfach »Informatik«.

In diesem hier kurz geschilderten Umfeld bildungspolitischer Diskussionen über eine informatische Bildung wurde die erste Tagung zum Thema »Informatik und Schule« 1984 in Berlin geplant und durchgeführt. Dass eine solche Tagung seinerzeit genau das Bedürfnis der Handelnden traf, zeigten die politischen Redner und die Zahl der Teilnehmerinnen und Teilnehmer: Der damalige KMK-Präsident und Kultusminister Niedersachsens Georg-Berndt Oschatz und die Berliner Schulsenatorin jener Zeit, Hanna-Renate Laurien, ließen es sich nicht nehmen, den 650 Anwesenden ihre Gedanken zum Computereinsatz in der Schule und zum Informatikunterricht vorzutragen.

Zugleich bot sich während der Tagung die Gelegenheit, über die im Tagungsband dokumentierten bereits beschlossenen GI-Empfehlungen, aber auch über Empfehlungen der GI zur Informatik in Schule und Ausbildung zu diskutieren, die zu dieser Zeit erst in ihren Arbeitsfassungen vorlagen [vgl. AH84, S. 337 ff.]. Wie weiter unten noch gezeigt werden wird, ist diese Tradition, über Arbeitsfassungen von Empfehlungen auf INFOS-Tagungen mit den Teilnehmerinnen und Teilnehmern zu diskutieren, unter anderem bei der Entstehung der »Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I« wieder aufgegriffen worden.

Heutzutage ist Informatik etwas Selbstverständliches geworden – allerdings immer noch nicht der Informatikunterricht. Von der Armbanduhr über die Waschmaschine bis zum Fernseher im Wohnzimmer ist jeder Haushalt mit Computern ausgestattet – ganz zu schweigen von Handys, Laptops und PCs. In all diesen für selbstverständlich gehaltenen Gegenständen stecken die Ideen, für die die Wissenschaft Informatik die Basis darstellt. Doch diese Ubiquität, diese Allgegenwart der Informatik ist den wenigsten Menschen bewusst. Diese Allgegenwart hat die Informatik unsichtbar werden lassen. Und wer

⁶ [BLK87] Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung: Gesamtkonzept für die informationstechnische Bildung. Reihe »Materialien zur Bildungsplanung und zur Forschungsförderung«, Heft 16. BLK-Geschäftsstelle, Bonn, 1987.

irgendein Textverarbeitungsprogramm meint bedienen zu können, seine E-Mails lesen und absenden und sich im Internet einigermaßen zurechtfinden kann, will auch gar nichts von Informatik wissen. Fernsehen könne man auch ohne Physikstudium, ist die gängige Meinung.

Während der 10. Fachtagung in Garching bei München begann deshalb – ausgelöst durch die Vorträge von Steffen Friedrich über »Informatik und PISA – vom Wehe zum Wohl der Schulinformatik« [in: Hu03, S. 133 ff.] und von Hermann Puhmann über »Informatische Literalität nach dem PISA-Muster« [in: Hu03, S. 145 ff.] erneut eine Diskussion um den Kern des Informatikunterrichts. In der Folgezeit bildete sich ein Arbeitskreis, der sich das Ziel setzte, Bildungsstandards für das Schulfach Informatik in der Sekundarstufe I zu formulieren. Bei Tagungen in Königstein (Sachsen) und in Eschenbach (Mittelfranken) wurden diese Bildungsstandards weiter ausgearbeitet. Zwischenergebnisse wurden wiederum auf den folgenden INFOS-Tagungen in Dresden⁷ 2005 und Siegen⁸ 2007 präsentiert und kritisch mit den Teilnehmerinnen und Teilnehmern diskutiert. Hier zeigte sich wieder einmal, welche positiven Kommunikationsmöglichkeiten INFOS-Tagungen bieten. Letztlich wurden die Standards im Januar 2008 vom Präsidium der Gesellschaft für Informatik verabschiedet und veröffentlicht⁹. Welche Wirkungen sie bislang in der bildungspolitischen Diskussion der einzelnen Bundesländer und welchen Einfluss sie möglicherweise auf die Formulierung neuer Kern- und Rahmencurricula gehabt haben und noch haben werden, wird sich unter anderem in den Gesprächen und Diskursen auf der nunmehr wieder in Berlin stattfindenden INFOS zeigen.

Ziel des Arbeitskreises war, mit den Bildungsstandards »eine zeitgemäße und fachlich substantielle informatische Bildung in den Schulen zu befördern« [AKBSI08, S. V]. Auch der Anspruch der Bildungsstandards ist nicht gerade gering, werden sie doch von der Vision getragen, »dass informatisch gebildete Menschen alle informatischen Probleme, die ihnen in ihrem Leben begegnen werden, mit Selbstvertrauen anpacken und selbstständig allein oder im Team bewältigen können. Und die Lehrenden helfen den Lernenden dabei, ihre Kompetenz zum Lösen solcher Probleme einzusetzen, zu vertiefen und auszubauen« [AKBSI08, S. 1]. Dass diese Ziele auch ein Leitmotiv der Tagungsreihe »Informatik und Schule« sein könnten, versteht sich von selbst.

25 Jahre – ein Vierteljahrhundert – sind für die Computerentwicklung nahezu eine Ewigkeit. Doch wer sich mit Bildungsprozessen auseinandersetzt, muss in solchen Zeiträumen denken. Denn informatische Bildungsziele und -inhalte folgen nicht dem Moore⁹

⁷ [Fr05] Friedrich, S. (Hrsg.): Unterrichtskonzepte für informatische Bildung – INFOS 2005 – 11. GI-Fachtagung Informatik und Schule, 28.–30. September 2005 in Dresden. Reihe GI-Edition »Lecture Notes in Informatics«, Band P-60. Köllen Verlag, Bonn, 2005.

⁸ [Sc07] Schubert, S. (Hrsg.): Didaktik der Informatik in Theorie und Praxis – INFOS 2007 – 12. GI-Fachtagung Informatik und Schule, 19.–21. September 2007 in Siegen. Reihe GI-Edition »Lecture Notes in Informatics«, Band P-112. Köllen Verlag, Bonn, 2007.

⁹ [AKBSI08] Arbeitskreis »Bildungsstandards« der Gesellschaft für Informatik (Hrsg.): Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I. Empfehlungen der Gesellschaft für Informatik e. V. vom 24. Januar 2008. In: LOG IN, 28. Jg. (2008), Heft 150/151, Beilage.

schen Gesetz. Wer heute etwas in der Schule ändert, kann die volle Wirkung dessen erst in den nächsten Jahren oder gar Jahrzehnten erleben.

Deshalb ist es auch eine besondere Freude zu wissen, dass die Reihe der Tagungen zu »Informatik und Schule« nicht beendet ist, sondern die nächsten Tagungen – an der Westfälischen Wilhelms-Universität Münster und der Friedrich-Alexander-Universität Erlangen-Nürnberg – bereits in Planung sind.

Schließlich sei in diesem, aufgrund des 25-jährigen Jubiläums etwas ausführlicher gewordenen Vorwort zum Tagungsband noch allen denjenigen herzlich gedankt, die zum Gelingen dieser Tagung beigetragen haben. Trotz einiger widriger Umstände ist es allen Beteiligten gelungen, durch ihre Arbeit, ihre Beiträge, ihr Engagement einen Einblick in den aktuellen Stand der informatischen Bildung zu geben, der für alle Teilnehmerinnen und Teilnehmer dieser Tagung nutzbringend sein kann. In erster Linie möchte ich allen Autorinnen und Autoren für ihre Beiträge zu diesem Band und damit für das reichhaltige Angebot danken, das eine solche Tagung belebt. Ein besonderer Dank gilt darüber hinaus den Mitgliedern des Programmkomitees der INFOS 2009 sowie den Mitgliedern des Organisations- und Örtlichen Programmkomitees, ohne deren Einsatz eine solche Tagung nicht gelingen kann. Auch den Mitarbeiterinnen und Mitarbeitern der Gemeinsamen Einrichtung Datenverarbeitung und informatische Bildung (GEDiB) des Fachbereichs Erziehungswissenschaft und Psychologie der FU Berlin danke ich für Geduld und Einsatz, neben der üblichen Arbeit auch noch bei den Vorbereitungen und der Durchführung der Tagung mitzuwirken. Namentlich möchte ich aber vor allem Helga Wolf und Manfred Reimer danken, deren Engagement bei den Vorbereitungen zu dieser Tagung mir eine außergewöhnliche Hilfe war. Und im Besonderen möchte ich meinem Freund und Kollegen Ingo-Rüdiger Peters danken, ohne dessen unermüdlichen Einsatz bereits vor 25 Jahren die erste INFOS in keinem Fall so erfolgreich geworden wäre.

Berlin, im September 2009

Bernhard Koerber



13. GI-Fachtagung »Informatik und Schule«

Zukunft braucht Herkunft – 25 Jahre INFOS

Veranstaltendes Fachgremium der Gesellschaft für Informatik

Fachausschuss »Informatische Bildung in Schulen«

Programmkomitee

Bänsch, Christian (Berlin)
Breier, Norbert (Hamburg)
Büttner, Katrin (Heidenau)
Diethelm, Ira (Oldenburg)
Fothe, Michael (Jena)
Friedrich, Steffen (Dresden)
Hartmann, Werner (Bern)
Klaaßen, Monika (Sanitz)
Koerber, Bernhard (Berlin)
Micheuz, Peter (Klagenfurt)
Penon, Johann (Berlin)
Peters, Ingo-Rüdiger (Berlin)
Puhlmann, Hermann (Altdorf)
Röhner, Gerhard (Darmstadt)
Schubert, Sigrid (Siegen)
Schulte, Carsten (Berlin)
Thomas, Marco (Münster)

Organisationskomitee und Örtliches Programmkomitee

Alsleben, Erlend (Berlin)*
Ebner, Roland (Königs Wusterhausen)
Koerber, Bernhard (Berlin)
Koubek, Jochen (Berlin)*
Lösler, Thomas (Herzberg)*
Neumeyer, Steffen (Bernau)*
Penon, Johann (Berlin)
Peters, Ingo-Rüdiger (Berlin)
Romeike, Ralf (Potsdam)*
Schulte, Carsten (Berlin)
Schulze, Peter M. (Ludwigfelde)
Vollmost, Manfred (Ludwigfelde)
Witten, Helmut (Berlin)

* insbesondere für den Tag der Schulen in Berlin-
Brandenburg (9. Tagung der GI-Fachgruppe
»Informatik-Bildung in Berlin und Brandenburg«)

Inhaltsverzeichnis

<i>Sigrid Schubert:</i> Aus Unterrichtsbeispielen lernen – Informatikdidaktische Partnerschaften	13
<i>Jens Gallenbacher:</i> Abenteuer Informatik – »Informatik begreifen« wörtlich gemeint	28
 <i>Einstieg in die Informatik (Sekundarstufe I und II)</i>	
<i>Wolfgang Pohl, Kirsten Schlüter, Hans-Werner Hein:</i> Informatik-Biber: Informatik-Einstieg und mehr	38
<i>Ernestine Bischof, Roland T. Mittermeir:</i> Informatik-Schnuppereinheiten zur Positionierung des Unterrichtsfachs	50
<i>Katrin Büttner, Thomas Knapp, Steffen Friedrich:</i> Untersuchungen zu informatischen Kompetenzen in Sachsen – eine empirische Studie	62
<i>Kirstin Schwidrowski:</i> Beitrag der informatischen Bildung zu Schlüsselkompetenzen am Beispiel Internetworking	73
 <i>Einstieg in die Informatik (Primarstufe und Sekundarstufe I)</i>	
<i>Henry Herper, Volkmar Hinz:</i> Informatische Bildung im Primarbereich	74
<i>Rita Freudenberg:</i> Lernen mit Etoys	86
<i>Michael Weigend:</i> Algorithmik in der Grundschule	97
<i>Markus Weber, Bernhard Wiesner:</i> Informatische Konzepte mit Robotern vermitteln – Ein Unterrichtsprojekt für die Sekundarstufe I	109

*Forschungs- und Entwicklungsprojekte zur Didaktik und Methodik
des Informatikunterrichts*

Albrecht Ehlert, Carsten Schulte:

Unterschiede im Lernerfolg von Schülerinnen und Schülern in Abhängigkeit
von der zeitlichen Reihenfolge der Themen (OOP-First bzw. OOP-Later) 121

Christian Kollee, Peer Stechert, Sigrid Schubert:

Kompetenzentwicklung mit Informatiksystemen (KIS) 133

Matthias Heming:

Informatische Bildung mit Mobiltelefonen? Ein Forschungsbericht 134

Magnus Rabel, Reinhard Oldenburg:

Konzepte, Modelle und Projekte im Informatikunterricht –
Bewertungen und Erwartungen von Schülern und Studenten 146

Qualitätsentwicklung und Qualitätssicherung der informatischen Bildung

Torsten Brinda, David van de Water:

Wie gewinnt man Schülerinnen und Schüler für ein Informatikstudium?
Maßnahmen deutscher Hochschulen 157

Ulrich Kiesmüller:

Prozessbegleitende, automatisierte Identifizierung der Problemlösestrategien
von Lernenden bei Lösen algorithmischer Probleme 169

Kirsten Schlüter:

Eine Studie zu den Merkmalen der Aufgabenschwierigkeit
am Beispiel eines Informatik-Schülerwettbewerbs –
Erster Teil: Aufgabenklassifizierung 181

Wolf Spalteholz:

Was heißt hier Rechnernetze? 193

Informatische Bildung im Wandel der Zeit

Lothar Sack, Helmut Witten:

Zurück in die Zukunft?
Zur Geschichte der Rahmen(lehr)pläne Informatik Sekundarstufe II
in Berlin (West) 205

Maria Knobelsdorf:

Unterricht im Wandel? – Das Potenzial der Informatik 218

Elin-Birgit Berndt:
Integration statt Sahnehäubchen –
Die technologische Basis der Kulturtechniken hat sich verändert 230

Peter Micheuz:
Zahlen, Daten und Fakten zum Informatikunterricht
an den Gymnasien Österreichs 243

Didaktische und methodische Aspekte des Informatikunterrichts

Ralf Romeike:
Softwaretools für kreatives Lernen im Informatikunterricht 255

Christian Wach, Jens Gallenbacher:
Spielend sortieren mit Las Vegas Cardsort 256

Jochen Koubek, Carsten Schulte, Peter Schulze, Helmut Witten:
Informatik im Kontext (IniK) –
Ein integratives Unterrichtskonzept für den Informatikunterricht 268

Arno Pasternak, Jan Vahrenhold:
Rote Fäden und Kontextorientierung im Informatikunterricht 280

Michael Hielscher, Christian Wagenknecht:
Programming-Wiki: Online programmieren und kommentieren 281

Holger Roland:
Von der ersten Entitätsklasse zum Webshop –
Datenbanken in allgemein bildenden Schulen 293

Beat Döbeli Honegger, André Frey, Philippe Braxmeier:
Mit iLearnIT.ch spielerisch das Interesse an Informatik wecken 304

Diana Jurević, Nando Stöcklin, Werner Hartmann:
Informationskompetenz: ein Thema für den Informatikunterricht 316

Didaktische und methodische Konzepte zum Programmieren im Informatikunterricht

Lutz Kohl:
Die visuelle Programmiersprache Puck – Entwicklung, Erprobung, Reflexion 328

Michael Dohmen, Johannes Magenheim, Dieter Engbring:
Kreativer Einstieg in die Programmierung –
Alice im Informatik-Anfangsunterricht 329

Aspekte der Informatikgeschichte im Unterricht

Michael Fothe:

Adam Ries und das Linienrechnen –
ein historisches Thema für den Informatikunterricht 330

Jürgen Müller:

Computus – Der Weg eines Begriffs durch die Geschichte 340

Informatik in der Lehrerbildung

Ludger Humbert:

Informatikdidaktik – Einschätzung der Landschaft 353

Hanno Schauer:

Der Bologna-Prozess in der Lehrer-Ausbildung
aus dem Blickwinkel der Informatik –
Vorschläge zur Konsolidierung einer laufenden Reform 354

Visionen für die informatische Bildung

Carsten Schulte:

Dualitätsrekonstruktion als Hilfsmittel
zur Entwicklung und Planung von Informatikunterricht 355

Jochen Koubek:

Referenzprojekte für den Informatikunterricht –
Unterrichtsprojekte zwischen Selbst- und Fremdbestimmung 367

Autorenverzeichnis 368

Aus Unterrichtsbeispielen lernen – Informatikdidaktische Partnerschaften

Sigrid Schubert

Didaktik der Informatik und E-Learning, Universität Siegen
Hölderlinstraße 3, 57076 Siegen, sigrid.schubert@uni-siegen.de

Abstract: Die Partnerschaften zwischen Schule und Hochschule ermöglichten die Gestaltung und Überprüfung von Unterrichtsbeispielen zu Internetworking im Besonderen und zu Informatiksystemen im Allgemeinen. Das Konzept der Didaktischen Systeme und die Forschungsmethodik mit Intervention bewährten sich dabei. Mit ausgewählten Beispielen werden diese Unterrichtsprojekte vorgestellt und ihr Beitrag zur Theoriebildung in der Didaktik der Informatik diskutiert.

1 Motivation

Die Arbeitsgruppen „Didaktik der Informatik“ an der Hochschule suchen Praxispartner in der Schule. Solche informatikdidaktischen Partnerschaften sind unverzichtbar für die Praxisphasen in der Lehrerbildung. Konkreter Unterricht wird von Lehramtsstudierenden in Gruppenarbeit vorbereitet. Der Unterricht wird von den Betreuern aus Schule und Hochschule und der Studiengruppe beobachtet und ausgewertet. Für die Schule ist das eine anspruchsvolle Zusatzaufgabe, die von der Mitwirkung an der Betreuung bis zur Abstimmung der Unterrichtszeitplanung reicht. Umgekehrt wenden sich Schulen an die Hochschulgruppen mit sehr konkreten Erwartungen dazu, was die gemeinsamen Unterrichtsprojekte bewirken sollen. In unserem Fall wählten Informatiklehrende die informatikdidaktischen Partnerschaften, um gemeinsam Ideen zur Veränderung von Informatikunterricht umsetzen zu können. In aller Regel heißt das, die innovativen Unterrichtsbeispiele einschließlich der erforderlichen Unterrichtsmittel sind dafür selbst zu entwickeln. Studierende liefern hier eine wertvolle Unterstützung mit ihren Leistungen in Projektgruppen, Hausarbeiten und Diplomarbeiten. Das Einwerben von gemeinsamen Forschungsprojekten stellt eine weitere Unterstützungsmöglichkeit dar. 2005 und 2008 begann die Siegener Forschungsgruppe „Didaktik der Informatik und E-Learning“ jeweils ein solches DFG-Projekt. Im Folgenden werden drei ausgewählte Unterrichtsbeispiele vorgestellt. Internetworking steht als Begriff für die informatischen Grundlagen des Internet (z.B. Architektur, Protokolle). Dafür entwickelte S. Freischlad ein Didaktisches System. Da das Internet ein Informatiksystem ist, stehen die Arbeiten in Beziehung zu denen von P. Stechert, der für Informatiksysteme ein Unterrichtsmodell gestaltete, in dem er die Entwurfsmuster der Softwareentwicklung als Wissensrepräsentation zur Vernetzung von fundamentalen Ideen der Informatik angewendet. Beide Forscher diskutierten und erprobten ihre Konzepte in informatikdidaktischen Partnerschaften und etablierten Kompetenzentwicklung mit Informatiksystemen als Siegener Schwerpunkt.

2 Didaktisches System Internetworking

2.1 Vorgehensweise

Die DFG fördert von 2005 bis 2009 im Rahmen des DFG Sonderforschungsbereichs und Forschungskollegs 615 „Medienumbrüche“ an der Universität Siegen erstmals ein Projekt, das der Didaktik der Informatik zuzuordnen ist. Ziel war die Entwicklung und Erprobung von Unterricht für das Fach Informatik in der Sekundarstufe zu Strukturen, Kommunikationsbeziehungen und Informationssicherheit im Internet [FS09]. W. Kempf, Schulleiter und Informatiklehrer am Gymnasium Auf der Morgenröthe in Siegen, wurde 2006 nach der Durchführung des ersten gemeinsamen Unterrichtsprojekts von acht Wochen (vgl. Tabelle1) mit einem leitfadengestützten Interview zu seinen Beobachtungen befragt. „Am besten haben mir die Stunden gefallen mit dem E-Mail-Client, weil die Lernenden konkret sehen konnten, was auf dem E-Mail-Server abläuft. ... Der Lehrer wies darauf hin, dass der Gesamtumfang zwar in Ordnung war, dass es aber wichtig ist, einzelne grundlegende Prinzipien stärker zu betonen. So wurden nach seiner Einschätzung das Schichtenmodell und auch das Domain Name System in der ersten Erprobung nicht angemessen berücksichtigt.“ [St07], S. 69-70. Die Erfahrungen der ersten Erprobung führten nicht nur zur Überarbeitung der Unterrichtsstunden. Die Lernaufgaben wurden nach ihrem Beitrag zu den Lernelementen analysiert und nach Lernabschnitten klassifiziert [FS07]. Dabei zeigte sich der Mangel an Erkundungsaufgaben. Dafür wurde die lernförderliche Software Filius als Komponente des Didaktisches System Internetworking gestaltet. Didaktische Systeme sind ein flexibler Verbund der Komponenten Wissensstrukturen, Aufgabenklassen und lernförderliche Software [FS07].

Lernabschnitt	Lernelemente
Dienste im Netz der Netze	Client-Server-Prinzip, Aufbau des Internets, Schulrechnernetz
Paketversand	Vermittlung, IP-Adressen
Organisation von Verbindungen	(De-)Multiplexer (Ports), logische Verbindungen (Verbindungsaufbau und Verbindungsabbau, Segmentnummern)
Schichtenmodell des Internets	Schichtenmodell (Nachrichtenaustausch, Verbindungsaufbau, Vermittlung, Übertragung)
Die logische Struktur des Internets	Domain Name System
DNS-Spoofing	DNS-Abfragen, DNS-Caching
E-Mail-Protokolle	E-Mail-Protokolle SMTP und POP3
Nachrichtenübertragung im Internet	E-Mail-Übertragungsweg

Tabelle 1: Gliederung des Unterrichtsprojekts.

Die Vorgehensweise bei der Fachdidaktikforschung mit Unterrichtsintervention umfasst folgende Schritte: (1) Analyse von Bildungsempfehlungen, (2) Entwurf und Begrün-

dung des Grobkonzeptes „Didaktisches System Internetworking“, (3) Ausarbeitung des Feinkonzeptes des Unterrichtsprojektes einschließlich der erforderlichen Lehr-Lern-Materialien, (4) Beobachtung der Unterrichtsintervention, (5) Evaluation des Lehr-Lernprozesses und der Lehr-Lernmaterialien mit Auswirkung auf Theoriebildung und Bildungsmedien, (6) Publikation der empirisch erprobte Ergebnisse zum „Didaktischen System Internetworking“. Ein quasiexperimenteller Untersuchungsansatz mit Interventionsgruppe und Kontrollgruppe ist im Informatikunterricht der Sekundarstufe II oft nicht möglich, da er zwei Gruppen (mit je 20 bis 30 Lernenden) mit gleicher Zusammensetzung erfordert. In unserem Fall geht es nicht um den Vergleich von Unterrichtsvarianten, sondern um die Entwicklung von neuen Unterrichtsbeispielen. Für Internetworking waren dafür vier Jahre erforderlich. Die Unterrichterprobungen in 2006, 2007 und 2008 zeigten die Schwachstellen des Konzeptes und führten in intensiven Diskussionen mit den Lehrenden zur Überwindung der Mängel. Seit 2009 empfehlen die Partner von Schule und Hochschule den erreichten Stand anderen Lehrenden aufgrund der guten Ergebnisse bei Motivation und Lernerfolg.

2.2 Übungssituationen

Die EPA (Einheitlichen Prüfungsanforderungen) Informatik führt im Lern- und Prüfungsbereich „Interaktion mit und von Informatiksystemen“ auf: „Kommunikation zwischen Computern, Netze (z.B. einfaches Kommunikationsprotokoll, einfaches Schichtenmodell) [Ku04, S. 6].“ Dazu findet man für die schriftliche Prüfung zwei Aufgaben mit Lösungsskizzen für das Grundkursfach: 1.2.4 Netzwerke [Ku04, S. 47-49], 1.2.6 Zeitsynchronisation [Ku04, S. 53-56], eine Aufgabe für das Leistungsfach: 1.3.1 E-Mail-Adressen [Ku04, S. 62] und ein Aufgabenbeispiel für die mündliche Prüfung: 2.4 Suchmaschine [Ku04, S. 71]. Die Lösungsskizzen liefern Hilfestellung zum Schwierigkeitsgrad von Übungssituationen, da die Anforderungsbereiche ausgewiesen sind, z.B.:

- Teilaufgabe x: „Firewall: Als weitere Sicherheitsmaßnahme werden Firewalls verwendet. Beschreiben Sie, was mit Firewalls erreicht werden soll [Ku04, S. 48]“.
- Lösung x: „Firewall: Mit einer hardwaretechnisch oder softwaretechnisch ausgeführten Firewall sollen offene Ports (z.B. http, E-Mail, ...) beschränkt, die Paketarten (z.B. ICMP) kontrolliert und die Protokolle (TCP, UDP) überwacht werden. I: 2, II: 2, III: 0 [Ku04, S. 49]“.
- Teilaufgabe y: „Beschreiben Sie die Vor- und Nachteile verbindungsloser Protokolle am Beispiel der Internettelefonie [Ku04, S. 53]“.
- Lösung y: „Der Vorteil der Internettelefonie mit Hilfe eines verbindungslosen Protokolls liegt im günstigen Preis. Das einfache Protokoll vermeidet die für die Bestätigung von Paketen notwendigen Zeiten. ... Im Gegensatz zur traditionellen Telefonie muss keine permanente Leitung geschaltet sein. Nachteilig ist die mäßige Sprachqualität. Sie kann bei Verlust oder Verzögerung von Datenpaketen insbesondere bei Überlastungen leicht zustande kommen. Ein weiterer Nachteil liegt in der nicht abhörsicheren Übertragung über das Internet. I: 0, II: 4, III: 4 [Ku04, S. 55]“.

Schon diese beiden Teilaufgaben zeigen, dass die Autoren mit „einfaches Kommunikationsprotokoll, einfaches Schichtenmodell“ keine EPA-gerechten Prüfungsaufgaben formulieren konnten. Dagegen ist „Kommunikation zwischen Computern, Netze“ zu unspezifisch, um überhaupt etwas ausschließen zu können. Das unterstreicht die Notwendigkeit der informatikdidaktischen Partnerschaften bei der Konkretisierung.

Der Lehrplan der ACM (Association for Computing Machinery) empfiehlt für alle Lernenden der Sekundarstufe I: „The basic components of computer networks (servers, file protection, routing protocols for connection/communication, spoolers and queues, shared resources, and fault-tolerance) [ACM03, S. 12].“ Für die Umsetzung findet man lediglich einen Hinweis auf ein Firmenvideo. Als Problem erwies sich also die Vorbereitung geeigneter Übungssituationen für die Lernenden in der Sekundarstufe II. Anregungen für Lebensweltbezug, Beispiele und Aufgaben fanden wir in den Lehrbüchern Informatik 1-3 (vgl. [Fal04], [Hal07], [Hal08]). Informatik 1 kommt in der Jahrgangstufe 7 von bayrischen Gymnasien zum Einsatz. Dort wird Informatik einstündig im Rahmen des Fächerverbundes Natur und Technik unterrichtet. Einen Schwerpunkt bilden vernetzte Informationsstrukturen (Internet), Austausch von Information (E-Mail). Zum Client-Server-Prinzip findet man dort altersgerechte Fragen. Der Start gestaltete sich für unsere Lernenden in Jahrgangstufe 11/2 ohne Vorwissen zu Internetworking ebenso.

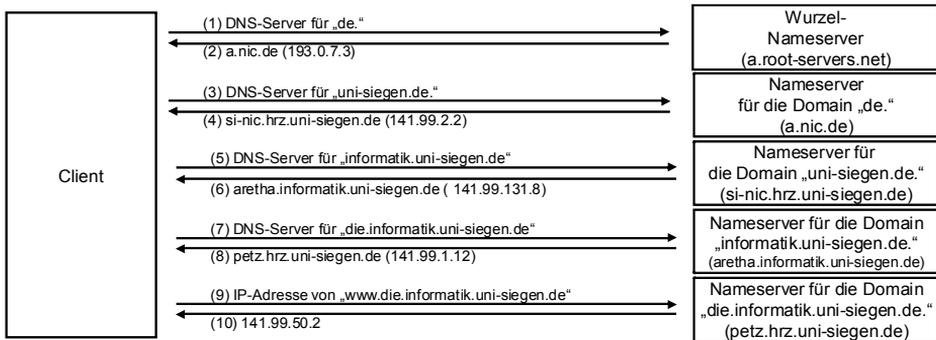


Abbildung 1: Kollaborationsdiagramm zur Auflösung eines Domainnamens.

Um die Regeln zur Auflösung eines Domainnamens zu strukturieren, konnte an das Vorwissen aus der objektorientierten Modellierung angeschlossen und mit Diagrammen gearbeitet werden (vgl. Abb. 1). Das Zustandsdiagramm veranschaulichte den Anmeldevorgang beim Mail-Server (vgl. Abb. 2). Dieser Ansatz erwies sich als erfolgreich.

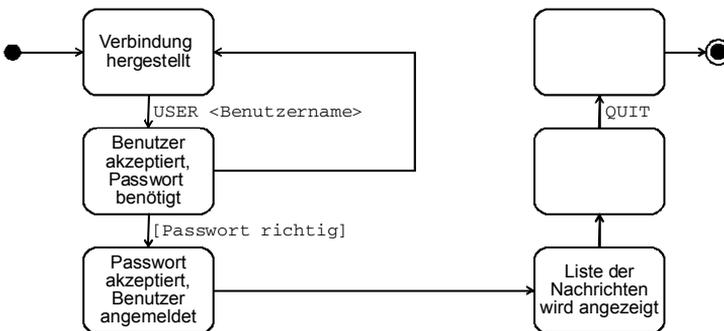


Abbildung 2: Zustandsdiagramm Mail-Server (POP3).

Er stößt aber an Grenzen. Er eignet sich nicht für die experimentellen Übungssituationen, die hier unverzichtbar sind, um das Systemverhalten zu erkunden. Dazu wurde Fili-

us, ein Unterrichtsmittel zum Simulieren verborgener Prozesse beim Internetworking, entwickelt (vgl. www.die.informatik.uni-siegen.de/pgfilius). Im Beispiel sollen die Lernenden verstehen, wie eine Paketfilter-Firewall zur Realisierung des Zugriffsschutzes eingesetzt werden kann. Es werden zwei Rechnernetze über einen Vermittlungsrechner miteinander verbunden (vgl. Abb. 3). In einem Rechnernetz steht ein Webserver zur Verfügung. In dem zweiten ist auf einem Rechner sowohl ein Webserver als auch ein DNS-Server installiert. Der Zugriff auf den zweiten Server kann durch eine Firewall kontrolliert werden. Auf den DNS-Server kann von überall zugegriffen werden, auf den Webserver nur innerhalb des lokalen Rechnernetzes. Die Lernenden vertiefen ihr Wissen über IP-Adressen und den Datenaustausch mit Ende-zu-Ende-Protokollen, indem Filterregeln für den Zugriff auf unterschiedliche Dienste erstellt werden.

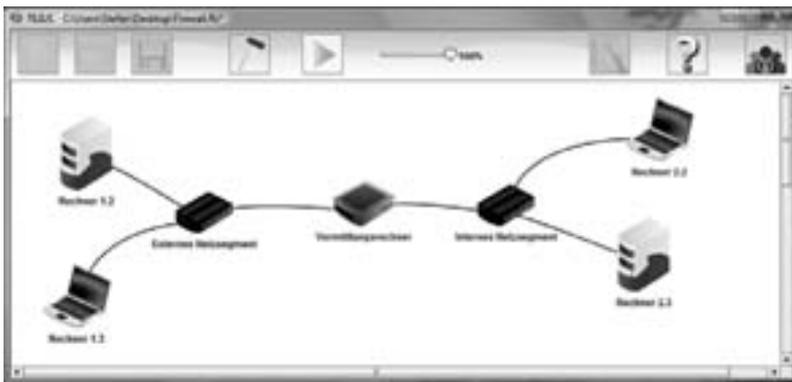


Abbildung 3: Rechnernetze mit Vermittlungsrechner.



Abbildung 4: Zugriffskontrolle mittels Firewall.

Mit Filius wurde im externen Netzsegment unter der Adresse www.extern.de der allgemein zugängliche Webserver auf Rechner 1.2 zur Verfügung gestellt und im internen Segment auf dem Rechner 2.3 ein DNS-Server und ein Webserver sowie eine Firewall installiert. Die Firewall ist so konfiguriert, dass der Port 53 für DNS-Anfragen für alle Rechner geöffnet ist. Der Port 80 für Webseitenabrufe ist nur für Rechner im gleichen Netzsegment geöffnet (vgl. Abb. 4).

3 Informatiksysteme und Entwurfsmuster

3.1 Verhalten erkunden

Parallel zu den Arbeiten zu Internetworking von Freischlad, entstand das Vorhaben von Stechert, Informatiksysteme im Unterricht nicht nur als Werkzeuge einzusetzen, sondern sie systematisch als Lerngegenstand anzueignen. Das schien unmöglich ohne Brücken zwischen dem nach außen beobachtbaren Systemverhalten und der intern verborgenen Systemarchitektur mit ihren Komponenten und Strukturen. Außerdem war erkennbar, dass die fundamentalen Ideen der Informatik [Sh93], eine fachdidaktische Vernetzung benötigen, um im Unterricht einen angemessenen Beitrag zur Kompetenzentwicklung mit Informatiksystemen zu leisten. Stechert setzte ausgewählte Entwurfsmuster der Softwaretechnik ein, um die Vernetzung von fundamentalen Ideen der Informatik damit zu veranschaulichen. Der Ansatz, ausgewählte Entwurfsmuster als Brücke von der Außenansicht zur Innenansicht von Systemen zu wählen, erforderte schulpraktische Erkundungen. 2006 und 2007 führt Stechert seine Unterrichtsprojekte am Fürst-Johann-Moritz-Gymnasium in Siegen durch. Seine These: „Die Unterteilung der Sichtweise in nach außen sichtbares Verhalten und innere Struktur von Informatiksystemen soll den Schülerinnen und Schülern einen Sichtenwechsel beim Verstehen des Programms Arztpraxis aufzeigen. Eine solche Unterteilung ist bei vielen Informatiksystemen hilfreich.“ erwies sich als tragfähig (vgl. Unterrichtsentwurf vom 17.10.2006, S. 4, www.die.informatik.uni-siegen.de/informatics-systems/index.html).

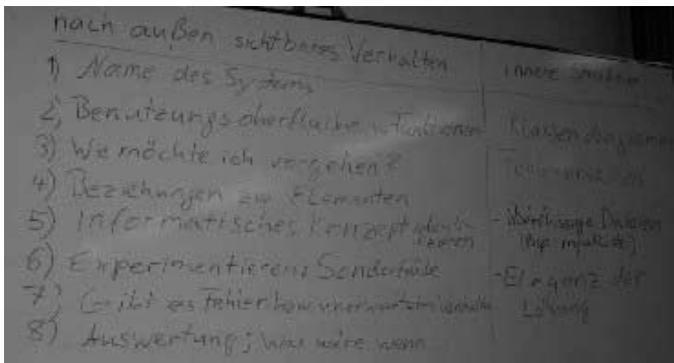


Abbildung 5: Ergebnis der Reflexion zur Vorgehensweise.

2006 setzte sich der Grundkurs (Jahrgangsstufe 12) aus 4 Schülerinnen und 18 Schülern zusammen, die objektorientierte Modellierung als Vorwissen mitbrachten. Als Beobachter beeindruckte mich, wie gut die zur Erkundung von Informatiksystemen empfohlene Vorgehensweise angenommen wurde (vgl. Abb. 5). Die Lehrenden betonten: „Ein zentrales Vorhaben war die Förderung eines vernetzten Denkens. Um das Verstehen von Informatiksystemen zu ermöglichen, soll dieses Denken aber nicht vom linear-analytischen Denken abgegrenzt, sondern mit diesem verknüpft werden. Eine Kombination dieser beiden Denkweisen im Unterricht erwies sich für die Schülerinnen und Schüler als schwierig [GK08, S. 119].“ Das zeigte sich auch im Übungsbedarf. Der vorbereitete Aufgabenvorrat war schnell aufgebraucht. So erwies es sich als hilfreich, ausgewählte Elemente der Lernsoftware Pattern Park im Unterrichtsprojekt zu erproben.

3.2 Strukturen erkennen

Die Auswertung mit den Informatiklehrenden ergab: „In der anschließenden Diskussion wurden problematische Aspekte der praktischen Umsetzung erläutert insbesondere konkrete Vorgehensweisen, um einen handlungsorientierten Unterricht zu ermöglichen [GK08, S. 118].“ Es war kein Problem, Experimente für das Erkunden des nach außen sichtbaren Systemverhaltens bereitzustellen. Die Brückenfunktion zum Erkennen der inneren Struktur können ausgewählte Entwurfsmuster aber nur übernehmen, wenn sie als Lernhilfe für die Wissensrepräsentation anwendbar sind. Für einen handlungsorientierten Zugang wurde deshalb die Lernsoftware Pattern Park von Studierenden (vgl. www.die.informatik.uni-siegen.de/pgpatternpark) entwickelt.

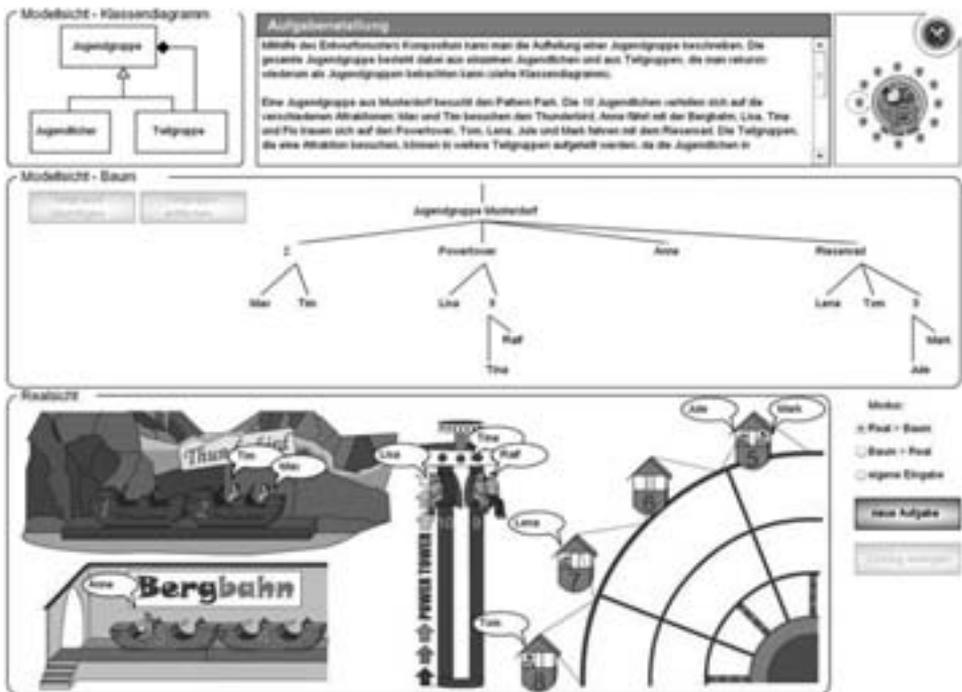


Abbildung 6: Sichtenwechsel zum Entwurfsmuster Kompositum.

Die Lernenden vollziehen den Sichtenwechsel zwischen Realsicht Freizeitpark und Modellsicht Baumstruktur (vgl. Abb. 6) in folgender Übung: Eine Jugendgruppe aus Musterdorf besucht den Pattern Park. Die zehn Jugendlichen verteilen sich auf die verschiedenen Attraktionen: Max und Tim besuchen den Thunderbird, Anne fährt mit der Bergbahn, Lisa, Tina und Flo trauen sich auf den Powertower, Tom, Lena, Jule und Mark fahren mit dem Riesennrad. Die Teilgruppen, die eine Attraktion besuchen, können in weitere Teilgruppen aufgeteilt werden, da die Jugendlichen in unterschiedlichen Wagen sitzen. So setzen sich zum Beispiel Jule und Mark gemeinsam in einen Wagen des Riesennrads, während Lena und Tom jeweils alleine in einem Wagen sitzen (siehe Realsicht). Die Datenstruktur Baum erleichterte hier den Zugang zur rekursiven Eigenschaft der Modellsicht des Entwurfsmusters Kompositum.

4 Kompetenzen zum Verstehen von Informatiksystemen

4.1 Lernförderliche Strukturmodelle

„Im Herbst 2007 erfolgte eine zweite Erprobung des Unterrichtsmodells zu den Themen ‚Die Zugriffskontrolle mit dem Proxymuster‘ und ‚Die Zustandsmodellierung von Systemverhalten mit dem Zustandsmuster‘ [GK08, S. 115].“ Die positive Bewertung der Informatiklehrenden zeigte zugleich die Richtung für die weitere Zusammenarbeit auf: „Die im Rahmen des Forschungsprojekts durchgeführten Unterrichtsreihen veränderten dieses Bild. Die Schülerinnen und Schüler haben die Erfahrung gemacht, dass Informatik mehr ist als nur programmieren. Das systematische Erkunden von Informatiksystemen, das im zweiten Halbjahr der Jgst. 12 und in der Jgst. 13 weiter vertieft wird, ermöglicht ein besseres Verständnis für die Vernetzung der zahlreichen Bereiche der Informatik ... [GK08, S. 119].“ Wir benötigten jetzt eine Einbettung der Kompetenzentwicklung mit Informatiksystemen in das Spiralcurriculum der Sekundarstufe II. Die erfolgreiche Anwendung der Entwurfsmuster war an Grenzen gestoßen. Das Abstraktionsniveau ist dabei sehr hoch. Entwurfsmuster eignen sich deshalb besonders als Strukturmodelle für Jahrgangsstufe 12/2. Welche Strukturmodelle sind für 11/2 und 12/1 geeignet? Von Oktober 2008 bis Januar 2009 führten wir dazu eine Unterrichtsintervention im Grundkurs 12/1 durch, in der in 24 Unterrichtsstunden mit anschließendem Test sowohl das Blockmodell des Von-Neumann-Rechners (VNR) als auch Schichtenmodelle (Ebenenmodell des Rechners, Internetschichtenmodell) zum Einsatz kamen [Kal09]. Die Auswertung ermöglichte eine Entscheidung zum Spiralcurriculum:

- Die erste Kompetenzvertiefung zum Verstehen von Informatiksystemen kann regelmäßig im Grundkurs 11/2 (erstmalig ab Mai 2009) in 4-5 Wochen erworben werden. Dabei wird ein Blockmodell als Lernhilfe eingesetzt.
- Die zweite Kompetenzvertiefung zum Verstehen von Informatiksystemen kann regelmäßig im Grundkurs 12/1 (erstmalig im Dezember 2009) in 4 Wochen erworben werden. Dabei werden ausgewählte Schichtenmodelle als Lernhilfe eingesetzt.
- Die dritte Kompetenzvertiefung zum Verstehen von Informatiksystemen kann regelmäßig im Grundkurs 12/2 (erstmalig ab April 2010) in 4 Wochen erworben werden. Dabei werden ausgewählte Entwurfsmuster als Lernhilfe eingesetzt.

Systemverständniskompetenz wird hier unterteilt in Kompetenz zur Interaktion mit Informatiksystemen (IS), Kompetenz zu Wirkprinzipien von IS, Kompetenz zu Wechselwirkungen zwischen IS, Individuum und Gesellschaft (z.B. Bildungsprozesse und Medien, Arbeitsprozesse und Werkzeuge, staatliche Verwaltungsprozesse und IS), Kompetenzen zum Experimentieren in der Informatik. Folgende Ergebnisse werden von Forschungsarbeiten in diesem Bereich erwartet:

- a. Studien zu Interaktion und Fachsprache; Studien zur Klassifikation von IS; Entwicklung von Handreichungen zu Aufbau und Durchführung von Experimenten;
- b. Studien zum Vernetzen fundamentaler Ideen in Block- und Schichtenmodellen, Entwurfsmustern, Architekturen usw.; Studien zum Sichtenwechsel (konkrete und abstrakte Repräsentationen);
- c. Studien zum Lesen von Spezifikationen, zur Aufteilung von IS in Subsysteme, zum Datenfluss zwischen Subsystemen, zur Bewertung von IS; Entwicklung von Vorgehensweisen von der Analyse der Komponenten zur Anwendung eines kognitiven Systemmodells.

4.2 Rollenspiele zum Blockmodell

Neben den Entscheidungen zur Integration in den Grundkurs bewegen uns auch viele Detailfragen. 2008 wählten wir zur Veranschaulichung der Funktionalität der Zentraleinheit ein Rollenspiel [Gi08], mit dem der Lernprozess gefördert werden konnte. Zwei Gründe bewegten uns zur Entwicklung eines etwas veränderten Rollenspiels:

1. Die Rollenbezeichnungen, z.B. „Schachtelzähler“, schöpfen das Potential zur Entwicklung der Fachsprache bei den Lernenden nicht aus.
2. Das Schachtelmodell behindert das Verständnis für den Speicher. Beim Auslesen von Befehlen und Daten wird eine Kopie übergeben, die Speicherzelle aber nicht geleert.

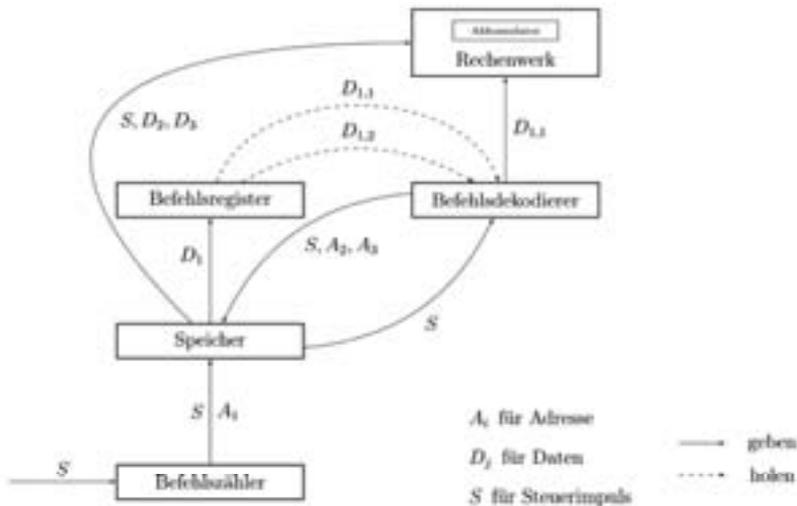


Abbildung 7: Rollenspiel zur Befehlsverarbeitung.

Das Rollenspiel beginnt mit dem Steuerimpuls durch die Lehrkraft (S) an die Rolle Befehlszähler. Die weiß, wo der nächste Befehl (A_1) zu finden ist. Sie notiert die Kopie der Adresse auf einem Zettel und gibt ihn zusammen mit dem Steuerimpuls (S) an die Rolle Speicher. Die liefert eine Kopie (Papier und Stift) des Befehls an die Rolle Befehlsregister (D_1), den Steuerimpuls (S) gibt sie aber an die Rolle Befehlsdekodierer. Die holt daraufhin die Teile des Befehls (mit Schere das Papier trennen) in Operation ($D_{1,1}$) und Adresse der Operanden ($D_{1,2}$) aus dem Befehlsregister. Danach gibt sie die Operation ($D_{1,1}$) an die Rolle Rechenwerk und den Steuerimpuls (S) zusammen mit den Adressen der beiden Operanden (A_2 und A_3) an die Rolle Speicher. Die gibt die beiden Operanden (D_2 und D_3) und den Steuerimpuls (S) an die Rolle Rechenwerk. Die führt die Operation (Papier und Stift) aus. Danach steht das Ergebnis im Akkumulator. Der Befehl wurde erfolgreich verarbeitet (vgl. Abb. 7). Wie immer beim Rollenspiel ist an Namensschilder zu denken. Farben helfen beim unterscheiden von Adressen, Daten und Steuerimpuls. Das Spiel sollte mit anderen Lernenden und einem anderen Befehl wiederholt werden, um die Erkenntnisse zur zentralen Steuerschleife zu festigen und zu vertiefen.

4.3 Entwicklung von Basiskompetenzen

Es existieren unterschiedliche Varianten zur Beschreibung der Kompetenzentwicklung im Informatikunterricht. Für die Sekundarstufe II geben die vier Kompetenzbereiche der EPA Informatik eine erste Orientierung:

- Erwerb und Strukturierung informatischer Kenntnisse,
- Kennen und Anwenden informatischer Methoden,
- Kommunizieren und Kooperieren,
- Anwenden informatischer Kenntnisse, Bewerten von Sachverhalten und Reflexion von Zusammenhängen.

Für mögliche Kompetenzstufen können die Anforderungsbereiche der EPA Informatik herangezogen werden:

- I: Wiedergeben, Beschreiben, Identifizieren, Verwenden, Übersetzen, Erweitern;
- II: Auswählen, Anordnen, Verarbeiten, Darstellen, Anwenden bekannter Verfahren;
- III: Auswahl und Anpassung geeigneter gelernter Methoden und Verfahren.

Erschwert wird die Anwendung der EPA Informatik auf das Verstehen von Informatiksystemen dadurch, dass der Inhaltsbereich „Wirkprinzipien von Informatiksystemen“ aufgeteilt wurde. So findet man „Grundsätzliche Funktionsweisen von Computersystemen (z.B. von-Neumann-Rechnermodell)“ unter Möglichkeiten und Grenzen informatischer Verfahren und „Kommunikation zwischen Computern, Netze (z.B. einfaches Kommunikationsprotokoll, einfaches Schichtenmodell) unter Interaktion mit und von Informatiksystemen [Ku04, S. 6]. Zusammenfassend bleibt unverständlich, warum die EPA Informatik in der Kompetenzentwicklung mit Informatiksystemen unter den Empfehlungen für die Sekundarstufe I bleibt [GI08]. Da wir bei den Lernenden in unseren Grundkursen kein Informatikvorwissen aus der Sekundarstufe I voraussetzen können, sind die dort zu erwerbenden Kompetenzen Bestandteil unserer Unterrichtsprojekte. Wir finden dort im Inhaltsbereich Informatiksysteme [GI08, S. 37-40] mehr Anregung als in der EPA Informatik, z.B.:

- verstehen die Grundlagen des Aufbaus von Informatiksystemen,
- erschließen sich weitere Informatiksysteme,
- klassifizieren Hardware und Software,
- reagieren angemessen auf Meldungen des Betriebssystems.

Interessant sind in diesem Zusammenhang auch die Vorüberlegungen zu Bildungsstandards Informatik für die Sekundarstufe II [Fo08].

Da wir mit dem Informatikunterricht nicht warten können, bis ein empirisch fundiertes Kompetenzmodell für Systemverständnis im Informatikunterricht der Sekundarstufe II entwickelt wurde, orientieren wir uns an dem Unterrichtsmodell von Stechert [St09, 190ff]. Es ermöglicht uns, die von ihm erprobten Lernphasen zu den Basiskompetenzen anzuwenden:

- S_A: Verständnis für die wesentlichen Aspekte des nach außen sichtbaren Verhaltens;
- S_{AB}: Verständnis für die Wechselwirkungen zwischen dem nach außen sichtbaren Verhalten und der inneren Struktur;
- S_B: Verständnis für die wesentlichen Aspekte der inneren Struktur, die auf fundamentalen Ideen der Informatik als Wirkprinzipien basieren (z.B. mit Hilfe von Strukturmodellen wie Blockmodellen, Schichtenmodellen, Entwurfsmustern);

- S_{BC} : Verständnis für die Wechselwirkungen zwischen der inneren Struktur und ausgewählten Implementierungsaspekten;
- S_C : Verständnis für die ausgewählten Implementierungsaspekte zur Erstellung einer konkreten Realisierung;
- S_{AC} : Verständnis für die Wechselwirkungen zwischen dem nach außen sichtbaren Verhalten und ausgewählten Implementierungsaspekten (z.B. durch systematisches Testen).

Im ersten Schritt konzentrieren wir uns auf vier der sechs Lernphasen, und zwar S_A , S_{AB} , S_B , S_{AC} . Davon wiederum ist S_{AB} die entscheidende. S_A und S_B bilden dazu die Erweiterung und ermöglichen den Sichtenwechsel. S_{AC} wiederum weist interessante fachdidaktische Parallelen zu S_{AB} auf, die bisher beim Erlernen der informatischen Modellierung und Programmierung unterschätzt wurden. Die folgenden Ergebnisse konnten bisher zu den Basiskompetenzen mit Hilfe des Blockmodells zum Von-Neumann-Rechner im Grundkurs 12/1 beobachtet werden (vgl. Tab. 2). Im laufenden Unterrichtsprojekt (ab Mai 2009) geht es um die Erkundung dazu im Grundkurs 11/2.

	Kognitive Kompetenzfacetten	Nicht-kognitive Kompetenzfacetten
	Die Schülerinnen und Schüler ...	
S_A	wissen, dass mehrere Programme gleichzeitig ausgeführt werden können	haben die Erwartungshaltung, dass mehrere Programme ohne Einschränkungen ausgeführt werden können
S_{AB}	können das Systemverhalten des VNR erklären und wissen, dass der VNR-Flaschenhals das Systemverhalten beeinflusst	sind bereit, von Black-Box-Vorstellungen (Gefahr der Computergläubigkeit) zu rationalen Strukturmodellen zu wechseln
S_B	können die zentrale Steuerschleife erklären (dynamisch Sicht)	sind motiviert, von einer statischen zu einer dynamische Sicht zu wechseln
S_{AC}	können einen Fehler im Systemverhalten auf das Dekodieren eines falschen Befehls aufgrund eines Speicherfehlers erklären	sind bereit, in Teamarbeit eine Fehlerursache zu analysieren

Tabelle 2: Basiskompetenzen mit Hilfe des Blockmodells zum Von-Neumann-Rechner.

Typisch für die informatikdidaktische Partnerschaft ist die gemeinsame Entwicklung und Erprobung von Unterrichtsmitteln. Die Informatiklehrenden wirken als Auftraggeber mit und beschreiben die Unterrichtsmittel, die sie zur Förderung der Handlungsorientierung einsetzen möchten. An der Universität werden diese Unterrichtsmittel dann von Studierenden gestaltet und bei Bedarf überarbeitet. So entstand zum Beispiel ein Applet zu einem Schichtenmodell, hier dem Ebenenmodell des Rechners (vgl. Abb. 8), das den Sichtenwechsel fördern soll. Die Lernenden wenden den Taschenrechner an (Anwendungssicht). Bei Eingabe von Operanden und Operation wird der Pseudocode (Ebene der problemorientierten Sprachen) ergänzt und der Assemblercode (Assemblerebene) erzeugt. Die Verarbeitung des Assemblerprogramms mit Hilfe der Hardware durch die Register und Arithmetik-Logik-Einheit (Ebene der Hardware) kann durch die Lernenden

wiederholt beobachtet werden.

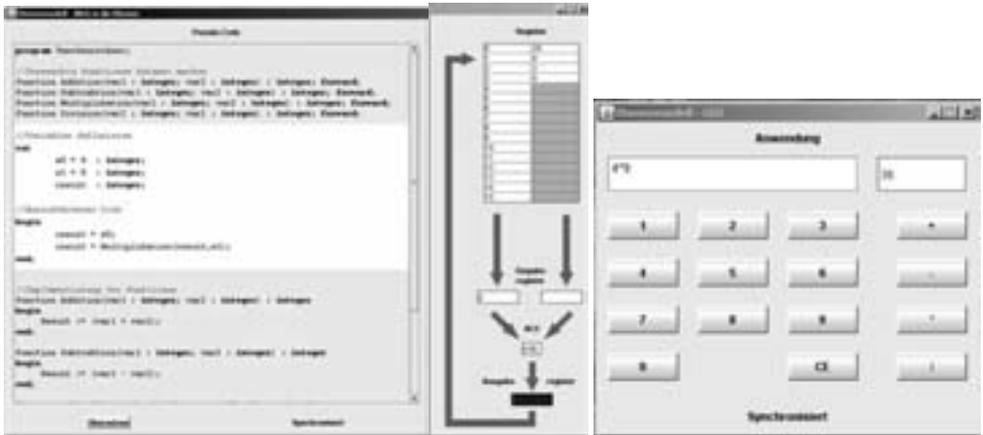


Abbildung 8: Schichtenmodell (Ebenenmodell) des Rechners.

Solche Unterrichtsmittel liegen mit Filius (vgl. Abb. 3 und 4) auch für das Vierschichtenmodell (Anwendung, Transport, Internet, Subnetz zur Datenübertragung) der Internet-Technologie vor, das die Lernenden handelnd erkunden können. Folgende Basis-kompetenzen, die mit Hilfe von Schichtenmodellen im Grundkurs 12/1 erworben wurden, sind exemplarisch für Internetworking beschrieben (vgl. Tab. 3).

	Kognitive Kompetenzfacetten	Nicht-kognitive Kompetenzfacetten
	Die Schülerinnen und Schüler ...	
S _A	tauschen über das Internet Daten aus	erwarten eine zuverlässige Datenübertragung
S _{AB}	können die Datenübertragung am Beispiel E-Mail anhand des Client-Server-Prinzips erklären	sind motiviert, sicherheitskritische Bereiche zu suchen
S _B	können die Datenübertragung mit einem Interaktionsdiagramm darstellen	sind bereit, den Kontext der Datenübertragung durch Anwendungsfälle einzubeziehen
S _{AC}	können die Datenübertragung systematisch nachverfolgen und Fehler auf Protokollebene erklären	nutzen ihr Wissen über Protokolle, um sich für mehr Datenschutz zu engagieren

Tabelle 3: Basiskompetenzen mit Hilfe des Schichtenmodells der Internet-Technologie.

Wir sprechen von Kompetenzfacetten (vgl. Tab. 2-4), um den beobachteten Lernerfolg in den Unterrichtsprojekten zu beschreiben, der an lebensweltliche Problemstellungen gebunden ist. Damit grenzen wir uns vom langwierigen Kompetenzerwerb ab, der erst nach mehreren Schuljahren ermittelt werden kann. Und wir warten den Einsatz von

Kompetenzmessinstrumenten ab, die zurzeit interdisziplinär entwickelt werden (vgl. Abschnitt 5).

	Kognitive Kompetenzfacetten	Nicht-kognitive Kompetenzfacetten
	Die Schülerinnen und Schüler ...	
S _A	kennen die Regeln der Zugriffskontrolle aus dem Schulrechnernetz	erwarten, dass auf persönliche Daten nur von einer berechtigten Person zugegriffen werden kann
S _{AB}	können die Realisierung der Zugriffskontrolle eines Systems mittels Stellvertreterobjekt erklären	sind motiviert, sicherheitskritische Bereiche zu suchen
S _B	können Zugriffskontrolle mit Klassendiagramm (Entwurfsmuster Proxy) und Sequenzdiagramm darstellen	sind bereit, in Teamarbeit die Struktur eines Programms zur Zugriffskontrolle zu analysieren
S _{AC}	sind in der Lage, bei gegebenem Quellcode die Zugriffsrechte zu modifizieren	können sich in die Sicherheitsbedürfnisse unterschiedlicher Benutzergruppen versetzen und daraus Anforderungen ableiten

Tabelle 4: Basiskompetenzen mit Hilfe des Entwurfsmusters Proxy.

Aus den Unterrichterprobungen von Stechert [St09], die ohne informatikdidaktische Partnerschaften nicht möglich gewesen wären, stammen die Beobachtungen der folgenden Basiskompetenzen, die mit Hilfe von Entwurfsmustern im Grundkurs 12 erworben wurden (vgl. Tab. 4). Betont wird in diesem Zusammenhang, dass es zu keinem Zeitpunkt darum ging, die Entwurfsmuster der Softwaretechnik zum vordergründigen Lerngegenstand der informatischen Bildung zu machen. Das würden unsere Partner in den Schulen nicht unterstützen. Vielmehr war es die Suche nach lernförderlichen Strukturmodellen auch im Bereich der Entwurfsmuster erfolgreich. Proxy- und Zustandsmuster bewährten sich dabei besonders, da sie den erforderlichen Lebensweltbezug sichern.

Die Entwicklung der informatischen Fachsprache erwies sich in allen Unterrichtsprojekten als kompliziert. Vermutet wird, dass diese Förderung mit Beginn der Sekundarstufe II zu spät einsetzt. Die Informatiklehrenden vermissen die Lehrbuchkultur, die sie aus ihrem zweiten Fach kennen. Damit fehlt ein Konsens zum Kernwortschatz der informatischen Fachsprache je Jahrgangsstufe. Unterschätzt werden ganz offensichtlich auch die Aufgaben aus dem Anforderungsbereich I (Sachverhalte und Methoden reproduzieren) und die damit verbundene Vorbildrolle der Lehrperson. Lehrende müssen mit großer Geduld Erklärungen wiederholen. Über das mehrfache Hören und Lesen können die Lernenden das aktive Anwenden der Fachwörter anbahnen: an Gesprächen teilnehmen, zusammenhängendes Sprechen und Schreiben in unterschiedlichen Kontexten. Ohne Zeit zum Sprechen und für Diskussionen (Erklären, Begründen, Bewerten) wird das Angebahnte aber nicht gefestigt. Solche Phasen fehlen häufig. Der Informatikunterricht scheint „in Eile“ voranzuschreiten, bleibt aber sprachlich auf einem Niveau, das für diese Jahrgangsstufen inakzeptabel ist.

5 Ausblick

Künftige Promotionsvorhaben in der Didaktik der Informatik sollten vermeiden, dass die forschende Person zugleich auch die Lehrperson in den empirischen Studien ist. Es kommt zu einer Überbelastung in der Unterrichtsbeobachtung, die kritische Reflexion erschweren kann. Informatikdidaktische Partnerschaften ermöglichen diese personelle Trennung und bringen Forschungsteams mit sich ergänzendem Expertenwissen zusammen.

Das Siegener Team wird die informatikdidaktischen Partnerschaften fortsetzen und pflegen. Der Schwerpunkt für 2010 liegt auf dem Messen von Kompetenzen zu Informatiksystemen. Mit dem Beginn neuer Promotionsvorhaben werden aber auch neue Unterrichtsmodelle in den Blickpunkt rücken.

Informatische Bildung zu Internetworking wird sich verändern. Das traditionelle Internet wird zum „Internet der Dinge“ mit dem Schwerpunkt eingebettete Mikrosysteme (EMS). Vernetzungsmöglichkeiten und Anbindung an verschiedene Übertragungsstandards, z.B. EDGE, sowie der Aufbau von Adhoc-Netzen, z.B. über Bluetooth, erweitern gegenwärtig die Infrastrukturen, wodurch zusätzliche Möglichkeiten geschaffen werden. Dazu fehlen erprobte Unterrichtskonzepte. Das könnte eine Entwicklungsrichtung unserer informatikdidaktischen Partnerschaften werden.

J. Magenheim, N. Schaper und S. Schubert leiten seit 2008 das DFG-Projekt „Entwicklung von qualitativen und quantitativen Messverfahren zu Lehr-Lern-Prozessen für Modellierung und Systemverständnis in der Informatik“. In diesem Projekt arbeiten Informatiker, Psychologen und Informatiklehrer aus Siegen und Paderborn an einem empirisch fundierten Kompetenzmodell für den Informatikunterricht der Sekundarstufe II. Die ersten Forschungsergebnisse wurden publiziert [Kal09a], [Kal09b], [Nal09]. Mit dem Vorstellen der entwickelten Messverfahren kann 2010 gerechnet werden.

Dank an die Partner

Seit 2002 wurden in Siegen viele Lehramtstudierende und Forschende von den Informatiklehrenden Wolfgang Kempf, Hartmut Koch, Werner Eling, Dr. Hans Schönwald in informatikdidaktischen Partnerschaften betreut. Unsere Absolventinnen Milena Ganea und Referendarin Natja Scharz kamen in den letzten Jahren neu hinzu. Ihnen allen danke ich für die ausgezeichnete Zusammenarbeit.

Literaturverzeichnis

- [ACM03] Association for Computing Machinery (ACM) (ed.): A Model Curriculum for K-12 Computer Science. 2003, www.csta.acm.org/Curriculum/sub/CurrFiles/K-12ModelCurr2ndEd.pdf, 1.5.2009.
- [Bal08] Brinda, T. et al. (Hrsg.): Didaktik der Informatik – Aktuelle Forschungsergebnisse. GI-Edition – Lecture Notes in Informatics (LNI), P-135, Bonn, 2008.

- [BI07] Benzie, D.; Iding, M. (eds.): Proceedings of IFIP-Conference on “Informatics, Mathematics and ICT (IMICT 2007): A golden triangle”. June 27-29, 2007, Boston, MA, www.die.informatik.uni-siegen.de/ifip-wg31/publications.html, 1.5.2009.
- [De07] Denning, P.: Great Principles of Computing. cs.gmu.edu/cne/pjd/GP, 1.5.2009.
- [Kal09a] Kollee, C. et al.: Kompetenzentwicklung mit Informatiksystemen – KIS. In: Koerber, B. (Hrsg.): Zukunft braucht Herkunft – 25 Jahre INFOS. GI-Edition – Lecture Notes in Informatics (LNI), Bonn, 2009, im Druck.
- [Kal09b] Kollee, C. et al.: Computer Science Education and Key Competencies, Fostering Key Competencies in Secondary Education by achieving Modeling Competencies and System Comprehension in Informatics. In: Proceedings of IFIP World Conference Computer in Education – WCCE 2009, Bento Gonçalves (Brazil) July 27-31, in press.
- [Ku04] Kultusministerkonferenz (Hrsg.): Einheitliche Prüfungsanforderungen Informatik. Beschluss der Kultusministerkonferenz vom 01.12.1989 i.d.F. vom 05.02.2004, http://www.kmk.org/fileadmin/veroeffentlichungen_beschluesse/1989/1989_12_01-EPA-Informatik.pdf, 1.5.2009.
- [Fal04] Frey, E. et al.: Informatik 1. Klett, Stuttgart 2004.
- [Fo08] Fothe, M.: Bildungsstandards Informatik für die Sekundarstufe II – Vorüberlegungen zur Entwicklung. In: [Bal08], S. 107-116.
- [Fr09] Freischlad, S.: Entwicklung und Erprobung des Didaktischen Systems Internetworking im Informatikunterricht. Dissertation, Universität Siegen, 2009.
- [FS07] Freischlad, S.; Schubert, S.: Towards High Quality Exercise Classes for Internetworking, In: [BI07].
- [GI08] Gesellschaft für Informatik (Hrsg.): Grundsätze und Standards für die Informatik in Schulen, Bildungsstandards Informatik für die Sekundarstufe I, 2008, www.informatikstandards.de, 1.5.2009.
- [Gi08] Gierhardt, H.: Informatik in der Oberstufe, www.oberstufeninformatik.de, 1.5.2009.
- [GK08] Ganea, M.; Koch H.: Kooperation zwischen dem Fürst-Johann-Moritz-Gymnasium und dem Institut „Didaktik der Informatik und E-Learning“ an der Universität Siegen. In: [ScSt08], S. 115-120.
- [Hal07] Hubwieser, P. et al.: Informatik 2. Klett, Stuttgart 2007.
- [Hal08] Hubwieser, P. et al.: Informatik 3. Klett, Stuttgart 2008.
- [Nal09] Nelles, W. et al.: Entwicklung eines Kompetenzrahmenmodells, Informatisches Modellieren und Systemverständnis. In: Informatik Spektrum (2009), im Druck.
- [Sal07] Schubert, S. et al.: Internetworking und Verstehen von Informatiksystemen. In: [St07], S. 65-74.
- [Sc07] Schubert, S. (Hrsg.): Didaktik der Informatik in Theorie und Praxis. GI-Edition – Lecture Notes in Informatics (LNI), P-112, Bonn, 2007, www.die.informatik.uni-siegen.de/e-publikationen/Tagungsbaende, 1.5.2009.
- [ScSh04] Schubert, S.; Schwill, A.: Didaktik der Informatik, Springer, Heidelberg 2004.
- [Sh93] Schwill, A.: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 1 (1993), S. 20-31.
- [St07] Stechert, P. (Hrsg.): Informatische Bildung in der Wissensgesellschaft. Reihe Medienwissenschaften, Bd. 6, Universitätsverlag Siegen – universi, Siegen 2007, www.die.informatik.uni-siegen.de/e-publikationen/Tagungsbaende, 1.5.2009.
- [St09] Stechert, P.: Fachdidaktische Diskussion von Informatiksystemen und der Kompetenzentwicklung im Informatikunterricht. Dissertation, Universität Siegen, 2009.
- [StSc07] Stechert, P.; Schubert, S.: A Strategy to Structure the Learning Process towards Understanding of Informatics Systems, In: [BI07].
- [ScSt08] Schubert, S.; Stechert, P. (Hrsg.): Bildungskonzepte für Internetworking und eingebettete Mikrosysteme. Siegen, Universitätsverlag Siegen – universi, 2008, www.die.informatik.uni-siegen.de/e-publikationen/Tagungsbaende, 1.5.2009.

Abenteuer Informatik – "Informatik begreifen" wörtlich gemeint

Dr.-Ing. Jens Gallenbacher

Didaktik der Informatik

Fachbereich Informatik, Technische Universität Darmstadt

Hochschulstraße 10

64285 Darmstadt

gallenba@informatik.tu-darmstadt.de

Abstract: Das Konzept "Abenteuer Informatik" zeigt den Kern der Wissenschaft bar von IT-Systemen und proprietären Anwendungen. "Begreifen" ist hier wörtlich gemeint. Der Artikel beschreibt, wie die Konzepte in der Schule umsetzbar sind, um neben dem "klassischen", konstruktiven Informatikunterricht noch einen Fächer weiterer Unterrichtsmethoden parat zu haben.

Neben dieser Unterrichtspraxis werden Möglichkeiten und erste Ergebnisse der Forschung dargestellt.

Motivation

Edsger W. Dijkstra wird der Satz zugeschrieben "In der Informatik geht es genauso wenig um Computer wie in der Astronomie um Teleskope". Trotzdem sieht der Alltag im Schulunterricht Informatik (wo dieser überhaupt stattfindet) oft anders aus: Computer dominieren – meistens schon physikalisch – das Klassenzimmer und auch den Unterricht. Auf diese Weise wird bereits unter Schülern das Bild der Informatik als "Computerlehre" geprägt – viele haben Berührungängste damit. Sei es, weil sie befürchten, gegenüber dominanten "Computerfreaks" ins Hintertreffen zu geraten, sei es, weil das Bild des vereinsamten Computerbedieners vor seinem Bildschirm in den Köpfen herumspukt. Auch politisch müssen wir aufpassen, nicht Informatik mit Informationstechnik oder Medienkompetenz in eine Schublade zu packen.

Moderne Computer stellen ein sehr komplexes und abstraktes System dar, was den Satz von Dijkstra noch verschärft: Während Schüler in Astronomie eine gute Chance haben, die Funktionsweise des Teleskops zu verstehen, ist dieses Ansinnen beim Computer mit seinen verschiedenen Hardware- und Softwarekomponenten utopisch. Karol, Cara, Greefoot & Co steuern hier in sehr geeigneter Weise gegen, indem sie eine vereinfachte Welt, das Spielfeld, bieten – allerdings müssen sich die Schüler darauf einlassen und akzeptieren, dass wesentliche Vorgänge "hinter den Kulissen" ablaufen: Eine wichtige Kompetenz, die der Informatikunterricht vermitteln soll ist die Fähigkeit zur Abstraktion. Gleichzeitig wird diese Kompetenz aber beim Umgang mit dem Unterrichtsmedium Computer bereits vorausgesetzt. Insbesondere in der Sekundarstufe I ist das nur schwer zu realisieren.

Informatik ohne Computer

Ganz andere Möglichkeiten bietet Informatikunterricht ohne Computer: Während in der Mathematik und den Naturwissenschaften einfach zu durchschauende Experimente "ohne Netz und doppelten Boden" eine lange Tradition haben, ist das Konzept in der Informatik sehr neu. In [Bell1995] kam ein erster Aufschlag mit dem erfolgreichen Konzept "Computer Science Unplugged". Bell entwickelt hier eine Informatik-Show für Kinder. Abenteuer Informatik greift die Idee auf, allerdings mit deutlich breiterer Zielgruppe und Experimenten in sehr unterschiedlichen Schwierigkeits- und Leistungsstufen.

Im Mittelpunkt steht dabei wörtlich das "Begreifen" der Thematik. Martin Wagenschein formulierte in [Wage1997]: "Ein Aspekt kann nur dann durchschaut werden, wenn man tätig dabei ist, wie er wird". Wenn das "Werden" dann allerdings ein Prozess ist, der im Wesentlichen in einer abstrakten Apparatur stattfindet, findet das Wagenschein'sche Durchschauen kaum statt.

Entgegen der vielfältig vorhandenen Technologie-Demonstrationen, über die man hauptsächlich aufgrund ihrer Komplexität erstaunt ist, benötigen wir Aha-Effekte, die durch ihre Einfachheit und Nachvollziehbarkeit hervorgerufen werden. Mit diesem Ziel im Sinn entstand 2006 zunächst das Buch "Abenteuer Informatik", das bereits 18 Monate später in zweiter Auflage erschien. Anhand eines Buches lassen sich die Konzepte allerdings nur schwierig evaluieren, daher folgte 2008 der Aufbau der Ausstellung "Abenteuer Informatik", die in Anlehnung an mobile Schülerlabore 15 Experimentierstationen bietet, anhand derer die Besucher wichtige Grundlagen der Informatik selbst begreifen können.

Gemeinsam ist allen Stationen, dass der Computer – wenn überhaupt – nur eine sehr untergeordnete Rolle spielt, etwa als Anzeigeelement. Die meisten Experimentieranordnungen sind auf einen Blick durchschaubar, weil sie etwa aus magnetischen Puzzleteilen bestehen oder bunte Schnüre gespannt werden müssen. Abbildung 1 zeigt einige Exponate.



Abbildung 1: Einige Exponate aus "Abenteuer Informatik"

Dynamische Programmierung

Demonstrieren möchte ich das am Beispiel der dynamischen Programmierung - einem Thema, das regelmäßig selbst Studierende der Informatik vor Verständnisprobleme stellt. Die Idee dahinter ist jedoch ganz einfach: *Um eine große Aufgabe zu lösen, ist es manchmal hilfreich, zunächst ganz kleine Aufgaben zu lösen und dann die Ergebnisse konsequent für die Gesamtlösung einzusetzen.* Man könnte das auch als "bottom-up"-Prinzip umschreiben.

Am besten wird das didaktisch anhand einer motivierenden Anwendung umgesetzt. Das gewählte Problem darf auf triviale Weise (z. B. mit Brute-force) nur sehr schwer zu lösen sein. Ein Lösungsweg muss gleichzeitig so übersichtlich darstellbar sein, dass man zumindest kleinere Aufgabenstellungen per Hand ohne zu viel Rechenarbeit und ohne Hilfsmittel lösen kann. Von den Algorithmen, die auf dynamischer Programmierung beruhen, etwa Bellman zur Generierung optimaler Binärer Suchbäume oder Floyd/Warshall zur Bestimmung kürzester Wege, erscheint die Lösung eines einfachen Rucksackproblems am meisten geeignet:

Gegeben sei ein Rucksack der Größe g_r , n Objekte $O_1..O_n$ mit den Größen $g_1..g_n$ und Werten $w_1..w_n$.

Gesucht ist eine Kombination π der Objekte, so dass für $O_{\pi_1}..O_{\pi_m}$ gilt

$$\sum_{x=1..m} w_{\pi_x} \text{ maximal und } \sum_{x=1..m} g_{\pi_x} \leq g_r$$

Auch wenn das Verständnis der formalen mathematischen Darstellung sicherlich eine wichtige Kompetenz ist, kann man für die Vermittlung der dynamischen Programmierung die Aufgabe auch deutlich "schöner" verpacken:

Professor Jonas ist endlich zum großen Schatz in der Pyramide vorgedrungen. Leider verrät ihm eine Inschrift, dass eine Falle aktiviert wurde: Er darf beim Verlassen nur soviel Gewicht haben wie beim Hineingehen. Außerdem stürzt danach alles zusammen und alle verbleibenden Schätze sind verloren. Glücklicherweise weiß Jonas, dass seine Ausrüstung genau 13 Kilo wiegt - wenn er sie zurücklässt, kann er das Gewicht in Schätzen mitnehmen. Professor Jonas kann außerdem für jeden Gegenstand ungefähr den Wert und das Gewicht bestimmen. Helfen Sie Professor Jonas: Versuchen Sie zunächst durch Probieren, einen möglichst hohen Schatzwert mit maximal 13 Kilogramm Gewicht zusammenzustellen. Die Waage zeigt Ihnen das aktuelle Gewicht an. Der Wert jedes einzelnen Schatzes steht rechts unter der entsprechenden Abbildung (in Tausend Dublonen).



Abbildung 2 zeigt die Umsetzung als Exponat.



Abbildung 2: Das Exponat in reger Benutzung

Das Beispiel ist so gewählt, dass man mit Mühe durch Ausprobieren auf die korrekte Lösung kommt und bereit ist, einen besseren Ansatz zu akzeptieren. Zunächst muss aber eine didaktisch geeignetere Repräsentation gefunden werden. Eine leichter zu durchschauende Sicht ergibt sich, wenn wir das Gewicht durch die ebenfalls lineare Länge eines Objekts ersetzen. Das Problem ist nun überschaubarer, aber immer noch gleich komplex. Abbildung 3 zeigt gegenübergestellt "reale" und "virtuelle" Objekte.



Abbildung 3: Reale und virtuelle Objekte

Die Schüler erkennen, dass man auch mit den abstrakteren Schätzen nicht um das konsequente Probieren herumkommt, um die Aufgabe zu lösen.

Als nächstes wird das Prinzip der dynamischen Programmierung auf dem einfachsten Level eingeführt: Man versucht, zunächst die kleinste, denkbare Aufgabe zu lösen und steigert sich dann langsam. Was ist die kleinste denkbare Aufgabe beim Rucksackproblem? Richtig: Man kann die Größe der Kiste reduzieren, aber auch die Anzahl der Schätze. Machen wir beides und fangen mit nur dem kleinsten Schatz und einer Kiste der Größe 0 an. Danach arbeitet man sich vor und füllt auch die nächstgrößeren Kisten mit dem kleinen Schatz auf. Im Exponat wird dann als Dialog zwischen Prof und Studentin die weitere Vorgehensweise erklärt:

"Für eine mit Schätzen bis zum momentanen Wert gefüllten Kiste gibt es bei der Hinzunahme eines weiteren Schatzes immer nur zwei Möglichkeiten:

- Die Kiste war vorher schon optimal gefüllt
- Die Kiste ist mit dem neuen Schatz optimal gefüllt

Trifft letzteres zu, können wir auch sagen, womit die Kiste sonst noch gefüllt ist – nämlich mit der optimal gefüllten Restkiste, die wir dank dynamischer Programmierung auch bereits bestimmt haben."

Die Schüler können sich mit magnetischen Plättchen zur Lösung "vorbereiten", wie Abbildung 4 zeigt.



Abbildung 4: 22 oder 23, welche Kiste mit Gewicht 6 ist günstiger?

Ein Computer kommt im Experiment nicht vor!

Weitere Beispiele für Exponate sind Las Vegas Cardsort, das in einem weiteren Tagungsbeitrag ausführlich besprochen wird, die Codierung nach Huffman, das Maximale Flussproblem, Zahlendarstellungen, Redundanz und Fehlerkorrektur usw.

Ausführlich werden die Exponate unter [Gall2008a] beschrieben, weitere Veröffentlichungen um Abenteuer Informatik sind [Gall2008b] und [Gall2009a].

Erfahrungen

Die Ausstellung wurde inzwischen an 7 Terminen insgesamt 19 Wochen lang gezeigt und von ca. 25.000 Menschen besucht, hauptsächlich Schülerinnen und Schüler mit oder ohne ihre Lehrer. Feedback wurde von den Besuchern durch einen Fragebogen, die Möglichkeit eines Eintrags im Gästebuch und durch direkte Interviews abgefragt.

Während der Ausstellung durften wir unsere Vorgaben bezüglich des Mindestalters deutlich nach unten korrigieren: Bereits Schüler der dritten Klasse konnten einige Exponate nicht nur "bespielen" sondern begriffen auch recht schnell den Kern der dort gezeigten Phänomene. Oft konnte ich beobachten, wie diese Kinder ihre Erkenntnisse dann sogar Mitschülern oder Eltern erstaunlich präzise erklärten.

Eine große "Abstimmung mit den Füßen" konnte während der Hobit, einer maßgeblichen Orientierungsveranstaltung für Ausbildung und Studium in der Rhein/Main-Region, beobachtet werden. Über 10.000 Schülerinnen und Schüler strömen drei Tage lang durch das Darmstadtium und 2009 konnten sie auf ca. 700 m² Fläche auch Abenteuer Informatik erleben. Auffällig war hier besonders der durchweg hohe Anteil von Schülerinnen (fast immer über 50%), die entgegen des Klischees hier sehr lange zubrachten und sich intensiv mit den Exponaten beschäftigten. Befragt, zeigten sie sich meistens erstaunt über die Inhalte, mit denen sie im Zusammenhang mit Informatik nicht gerechnet hätten. Auch dies legt nahe, dass Fächer wie Informatik, die für unsere informationsbetonte Gesellschaft und unsere auf Innovationen aufgebaute Wirtschaft äußerst wichtig sind, leider in der allgemeinbildenden Schule noch sehr schlecht repräsentiert werden und daher auf andere Weise ins "rechte Licht" gerückt werden müssen – nicht zuletzt, um die richtige Wahl eines Studienplatzes zu ermöglichen. Das setzt selbstverständlich auch die Universitäten in die Pflicht, sich entsprechend an einer Reform des Bildungssystems in diese Richtung zu beteiligen.

Lehrer, die an der begleitenden Fortbildung teilnahmen und dann ihre Klassen über die Ausstellung führten, konnten bisher einen nachhaltigen positiven Effekt auf die Erarbeitung entsprechender Themen im Unterricht beobachten. Dies ist allerdings bisher eher "anekdotisch" als empirisch belegt.

Weitere Entwicklung

Momentan ist die wissenschaftliche Auswertung der Frage in Arbeit, welchen Beitrag interaktive Exponate zu einer nachhaltigen Informatikausbildung leisten können. Hierzu setzen wir die Exponate für die Verwendung in definierten Schülerlaboren um, mit denen dann auch eine gezielte empirische Forschung möglich wird. Die gesammelten "anekdotischen" Erfahrungen sind hierfür sehr wichtig.

Da sich gezeigt hat, dass die Exponate eine gute Grundlage für verschiedene Aspekte des Informatikunterrichts darstellen, sollen diese auch für Lehrer einfacher einzubinden sein. In Seminaren an der Universität und Lehrerworkshops entstehen daher nach und nach Handreichungen vom einfachen Arbeitsblatt bis zum kompletten Unterrichtsentwurf. Hierfür ist die aktive Mitarbeit aller interessierten Lehrer nötig. Bitte stellen Sie ggf. Ihre eigenen Erfahrungen und Entwürfe im Forum allen anderen zur Verfügung.

Zu den Handreichungen gehören auch Unterrichtssets, die die Exponate in klein als Stationen für Lernwerkstätten und Schülerlabors enthalten, zusammen mit Anleitungen, die zum Selbstlernen geeignet sind.

Fazit

"Abenteuer Informatik" in Buchform und als Ausstellung zeigt, dass es nicht nur möglich ist, Informatik ohne Computer zu lehren, sondern dass entsprechende Konzepte in verschiedener Hinsicht sogar besser und nachhaltiger geeignet sind, den Kern der Wissenschaft zu vermitteln.

Informatik ohne Computer bewirkt außerdem eine Veränderung des "Bild vom Informatiker" in Öffentlichkeit und Schule, was wiederum ganz neue Interessenten aktiviert.

Literaturverzeichnis

[AKBS2008] – Arbeitskreis „Bildungsstandards“ der Gesellschaft für Informatik (Hrsg.) "Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I." Empfehlungen der Gesellschaft für Informatik e. V. vom 24. Januar 2008. In: LOG IN, 28. Jg. (2008), Heft 150/151, Beilage.

[Bell1995] Bell, T. C., Bensemann, G., and Witten, I. H. (1995) "Computer Science Unplugged: Capturing the interest of the uninterested" Proc NZ Computer Conference, Wellington, New Zealand, August

[Bell2005] Tim Bell, Ian H. Witten, Mike Fellows "Computer Science Unplugged-An enrichment and extension programme for primary-aged children, 2005. <http://csunplugged.com/>

[Gall2008] Jens Gallenbacher "Abenteuer Informatik: IT zum Anfassen", 2. erweiterte Auflage 2008; Springer/Spektrum Akademischer Verlag, Heidelberg; ISBN 978-3-8274-1926-2.

[Gall2008a] Jens Gallenbacher "Abenteuer Informatik", Beschreibung der Ausstellung auf <http://www.abenteuer-informatik.de/>

[Gall2008b] Jens Gallenbacher "Abenteuer Informatik – Informatik begreifen." In: LOG

IN, 28. Jg. (2008), Heft 152, S. 83–84.

[Gall2009] Christian Wach, Jens Gallenbacher "Spielend Sortieren mit Las Vegas Cardsort", hier mit INFOS Tagungsband

[Gall2009a] Jens Gallenbacher "Abenteuer Informatik – Das Affenpuzzle". In LOG IN, 154/155 Jg (2009) S 26-30

[Wage1997] Martin Wagenschein "Kinder auf dem Wege zur Physik"; Beltz, 2. Aufl. 1997, ISBN 978-3407290021

Informatik-Biber: Informatik-Einstieg und mehr

Wolfgang Pohl
Bundeswettbewerb Informatik, Ahrstr. 45, 53175 Bonn
pohl@bwinf.de

Kirsten Schlüter
Didaktik der Informatik
Universität Erlangen-Nürnberg

Hans-Werner Hein
Verlässliche IT-Systeme
Köln

Abstract: Mit dem Informatik-Biber gibt es seit 2006 einen neuen Informatik-Wettbewerb. Er ist der deutsche Partner der internationalen Wettbewerbs-Initiative „Bebras“. Der Informatik-Biber soll Jugendliche früh auf Informatik und ihre vielfältigen Aspekte aufmerksam machen und für die Beteiligung an Informatik-Wettbewerben gewinnen. Wie alle Bebras-Wettbewerbe wird er als Quiz durchgeführt, was an Inhalt und Form der Aufgaben besondere Anforderungen stellt. Diese werden anhand einer Beispielaufgabe beschrieben. Der Informatik-Biber konnte schnell hohe Teilnahmezahlen erreichen; die Zahlen aus 2008 werden näher beleuchtet. Schließlich wird besprochen, inwieweit die mit der Durchführung des Wettbewerbs verbundenen Ziele bereits erreicht sind.

1 Ursprünge und Entwicklung

Schülerwettbewerbe im MINT-Bereich wurden ursprünglich meist als Leistungswettbewerbe für Schülerinnen und Schüler mit besonderer Neigung angelegt. In Deutschland sind Jugend forscht, die Qualifikationswettbewerbe für die naturwissenschaftlichen Olympiaden oder die Bundeswettbewerbe Mathematik und Informatik Beispiele dafür. Breiten Zuspruch hingegen findet ein Wettbewerb, der 1978 in Australien ins Leben gerufen wurde, mittlerweile in vielen Ländern gleichartig durchgeführt wird und im Jahr 2008 allein in Deutschland mehr als 750.000 Teilnahmen verzeichnen konnte: *Känguru der Mathematik*. Ziel dieses Wettbewerbs ist „in allererster Linie die Popularisierung der Mathematik“. Die auf Papier im Multiple-Choice-Verfahren zu beantwortenden Aufgaben enthalten „kleine mathematische Probleme“; durch sie soll „Freude an (mathematischem) Denken und Arbeiten [...] geweckt bzw. unterstützt werden“.¹

Der Erfolg des Känguru-Wettbewerbs veranlasste die litauische Didaktikerin Valentina Dagiene, ein ähnliches Projekt für die Informatik zu initiieren. In 2004 fand der erste *Bebras* (dt.: Biber) Wettbewerb in Litauen statt [Da06]. Im Mai 2005 berieten im litauischen Pasvalys Vertreter mehrerer Länder über die Verbreitung der Bebras-Idee. Einen deutschen Vertreter hatte der Bundeswettbewerb Informatik (BWINF) entsandt. Die erste Ausrichtung eines deutschen *Informatik-Biber* wurde 2006 im Rahmen des Projekts *Ein-*

¹Alle Zitate von <http://www.mathe-kaenguru.de/wettbewerb/ziele/index.html> (Stand: 20.2.2009).

stieg Informatik möglich [PKH07], zunächst unter dem Namen *EI:SPIEL blitz*, aber von Anfang an in Form eines Online-Quiz.

Trotz kurzfristiger Anberaumung nahmen 2.186 Jugendliche aus 50 Schulen teil. Dieser Achtungserfolg ermutigte BWINF und Gesellschaft für Informatik (GI), das Projekt weiterzuführen. So wurde der Informatik-Biber auch in 2007 und 2008 ausgerichtet, mit knappen Mitteln und unter hohem persönlichen Einsatz vieler Beteiligter. Ab 2009 ist der Informatik-Biber als Einstiegsformat Bestandteil des Bundeswettbewerbs Informatik.

2 Zielsetzungen

Mit der Durchführung des Informatik-Biber verbinden die Veranstalter folgende Ziele:

Begegnung mit Informatik Informatik soll möglichst breit und in ihren vielfältigen Aspekten altersgerecht präsentiert werden. Indem Vorkenntnisse zur Lösung der Aufgaben nicht vorausgesetzt werden, kann die Teilnahme am Informatik-Biber eine erste, motivierende Gelegenheit zur Begegnung mit Informatik darstellen.

Werbung für Informatik Mit dem Informatik-Biber soll das Thema Informatik im schulischen Umfeld gestärkt werden. Die Wettbewerbsform soll hohe Teilnahmezahlen ermöglichen, die wiederum die Aufmerksamkeit einer breiten Öffentlichkeit auf das Themenfeld „Informatik und Schule“ lenken.

Frühe Ansprache Mit dem Informatik-Biber sollen Jugendliche schon früh auf Informatik aufmerksam werden. Damit besteht die Chance, Klischees und Vorurteile bzgl. der Informatik (etwa der Gleichsetzung von Informatik mit technischen Computerkenntnissen) rechtzeitig zu begegnen und die Informatik intellektuell attraktiv und herausfordernd zu präsentieren. Insbesondere Mädchen sollen in einem Alter erreicht werden, in dem das Interesse an naturwissenschaftlich-technischen Themen noch nicht durch Vorurteile und soziale Effekte behindert wird.

Heranführung an Informatik-Wettbewerbe Jugendliche, aber auch Lehrkräfte an den Schulen sollen an die Teilnahme an einem Informatik-Wettbewerb bzw. deren Betreuung herangeführt werden. Hier ist besonders der Bundeswettbewerb Informatik im Blickfeld; die BWINF-Ausrichter haben durchaus die Erfahrung gemacht, dass – trotz vorhandenen Potenzials – Jugendliche sich die Teilnahme bzw. Lehrkräfte sich deren Unterstützung nicht zutrauen. Als Ergänzung des bisherigen Angebots an Leistungswettbewerben (Bundeswettbewerb Informatik und Informatikolympiaden) soll der Informatik-Biber die von den BWINF-Verantwortlichen schon lange festgestellte Lücke bei der Ansprache von Jüngeren und Einsteigern schließen.

Informatik-Begabung identifizieren Der Informatik-Biber soll Jugendlichen eine einfach zugängliche Gelegenheit bieten, informatische Aufgaben zu bearbeiten und ihr Können in diesem Bereich zu entdecken. Lehrer werden in die Lage versetzt, als Beobachter in einer außerunterrichtlichen Situation wahrzunehmen, welche Aufgaben die Schüler besonders motivieren, welche ihnen besonders leicht oder besonders

schwer fallen. Sie können beobachten, welche Schüler sich über die Bearbeitungszeit hinaus mit den Aufgaben beschäftigen oder besonders erfolgreich sind. Weiter ermöglichen die anonymisierten Teilnahmedaten statistische Analysen der Beteiligung und des Erfolgs.

Internationale fachdidaktische Kooperation Unter den an Bebras beteiligten Ländern soll ein Abgleich der durchaus unterschiedlichen fachdidaktischen Schwerpunkte erreicht werden – soweit es die für einen solchen Wettbewerb relevanten Themengebiete der Informatik betrifft, sowie die Art und Weise der Aufgabenformulierung.

3 Aufgaben

3.1 Rahmenbedingungen

In den Bebras-Ländern werden jedes Jahr neue Vorschläge für Bebras-Aufgaben entwickelt. Auf einem internationalen Workshop werden die Vorschläge besprochen und in einer englischen Referenzversion ausgearbeitet. Aus dem so geschaffenen Aufgaben-Pool stellen die einzelnen Veranstalter ihre Wettbewerbe zusammen. Einige Aufgaben sind für alle Wettbewerbe obligatorisch; deren Anteil soll mittelfristig steigen. Ein internationales Board diskutiert neben organisatorischen auch fachwissenschaftliche Fragen der Wettbewerbsdurchführung wie etwa Inhaltskategorien für Aufgaben (s. Abschnitt 3.4).

Die Bebras-Wettbewerbe werden nach den folgenden international abgestimmten Regularien durchgeführt: Jeder Wettbewerb ist in drei Altersgruppen eingeteilt: *Benjamin*, *Junior* und *Senior*. In jeder Altersgruppe werden 15-20 Aufgaben gestellt. Die Aufgaben werden innerhalb der Altersgruppe zu gleichen Teilen je einer der Schwierigkeitsstufen *leicht*, *mittel* bzw. *schwer* zugeordnet. Je nach Schwierigkeitsstufe werden pro bearbeiteter Aufgabe unterschiedlich viele Punkte gutgeschrieben (bei richtiger Bearbeitung) bzw. abgezogen (bei falscher Bearbeitung). Die Zeit zur Bearbeitung der Aufgaben ist begrenzt. In einigen Ländern erfolgt die Bearbeitung über interaktive PDF-Dokumente. Die meisten Bebras-Wettbewerbe werden aber über Online-Wettbewerbssysteme durchgeführt.

Der Informatik-Biber war von Anfang an ein Online-Wettbewerb. Seine Altersgruppen umfassen die gesamte Sekundarstufe: Benjamin: Jahrgangsstufen 5-7, Junior: Jahrgangsstufen 8-10, Senior: Jahrgangsstufen 11-13. Im Informatik-Biber 2008 wurden in jeder Altersgruppe 18 Aufgaben gestellt, die maximale Bearbeitungszeit betrug 40 Minuten.

Aufgaben für den Informatik-Biber müssen auf die Bedingungen des Wettbewerbs zugeschnitten und wie Quiz-Aufgaben konzipiert werden. Durchschnittlich stehen pro Aufgabe nur gut zwei Minuten Zeit zum Durchlesen und Beantworten zur Verfügung. Auch die Zielsetzungen des Wettbewerbs müssen durch die Aufgaben unterstützt werden. Im folgenden werden die dazu erforderlichen Eigenschaften von Biber-Aufgaben am Beispiel der Aufgabe „Frecher Nager“ (s. Abbildung 1) diskutiert. Diese Aufgabe wurde im Informatik-Biber 2008 in der Altersgruppe Benjamin mit dem Schwierigkeitsgrad mittel und in der Altersgruppe Junior mit dem Schwierigkeitsgrad leicht eingesetzt.

Frecher Nager

Biber mögen es nicht, wenn überall Strommasten herumstehen. Wo immer sie einen finden, nagen sie ihn durch, so dass er umfällt. Die Leitungen, die mit diesem Mast verbunden sind, liegen dann alle nutzlos am Boden.

Der freche Biber Ben möchte jedes Leitungsnetz so schnell wie möglich zerstören. Alle Leitungen müssen am Boden liegen.

Bei welchem Leitungsnetz gelingt ihm das, wenn er nur zwei Masten durchnagt?

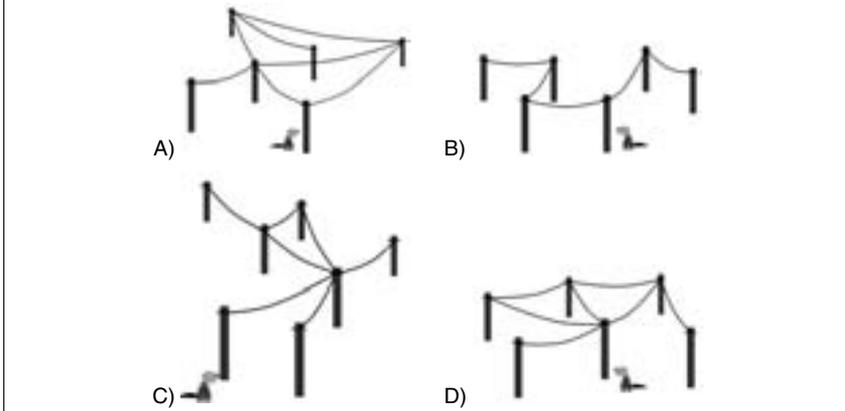


Abbildung 1: Aufgabe aus dem Informatik-Biber 2008

3.2 Typische Eigenschaften einer Biber-Aufgabe

Vorkenntnisse Um den Informatik-Biber für einen großen Teilnehmerkreis zu öffnen, sollen die Aufgaben möglichst keine Vorkenntnisse erfordern. In der Beispielaufgabe stellt der Einleitungstext alle zur Beantwortung der Frage nötigen Informationen zur Verfügung, weder Repräsentation noch Inhalt der Aufgabe stellen einen Anspruch an etwaige Vorkenntnisse.

Struktur Eine Biber-Aufgabe muss kurz sein, denn viel Zeit zum Lesen und Verstehen des Problems ist nicht vorhanden. Ein kurzer *Einleitungstext* beschreibt den inhaltlichen Hintergrund der Aufgabe und liefert eine Motivation für die Aufgabenstellung. Ein Teil der Hintergrund-Informationen kann auch grafisch vermittelt werden; das ist wegen des hohen Informationsgehalts von Bildern sehr sinnvoll. In der Beispielaufgabe wird im ersten Absatz das Szenario der Aufgabe beschrieben, während der zweite Absatz eine Zielsetzung liefert, auf die sich die spätere Fragestellung beziehen kann. Danach folgt die eigentliche *Fragestellung*, die in ihrer Formulierung zur abschließenden *Antwortmöglichkeit* passen muss. In der Beispielaufgabe werden Antwortalternativen angeboten, von denen eine auszuwählen ist (Multiple-Choice).

Fragestellung und Antwortform Die große Zahl an Teilnahmen macht eine automatisierte Auswertung erforderlich. Dafür muss die Menge der möglichen Antworten und insbesondere die der richtigen Antworten präzise definiert werden können. Die Fragestellung muss dementsprechend konkret ausfallen. Es werden also typischerweise Probleminstanzen behandelt, Fragen nach allgemeinen Problemlösungen sind kaum möglich. In der Beispielaufgabe werden vier mögliche Antworten zur Auswahl vorgegeben, genau eine dieser Antworten ist richtig. Multiple-Choice ist beim Informatik-Biber die bevorzugte Antwortform, in 2008 gab es sogar nur Multiple-Choice-Aufgaben. Alternativ erlaubt das Wettbewerbssystem des Informatik-Biber die freie Eingabe eines Antwortwertes.

3.3 Was macht eine gute Biber-Aufgabe aus?

Informatik-Bezug Eine gute Aufgabe hat einen klaren Bezug zu Informatik-Grundlagen oder auch zur Anwendung von Informatiksystemen. Damit ist sie deutlich als Informatik-Aufgabe erkennbar. Die Aufgabe kann so außerdem zur Veranschaulichung des zu Grunde liegenden Themas genutzt werden. Die Beispielaufgabe nimmt Bezug auf Graphentheorie, insbesondere die Bestimmung von Artikulationspunkten in einem Graphen: Nur Artikulationspunkte der in den Antwortalternativen implizit dargestellten Graphen kommen als zu durchnagende Masten in Frage. Artikulationspunkte deuten Schwachstellen in einem Netzwerk an, deren Ausfall das Netzwerk in Komponenten zerfallen lässt. Die Bestimmung von Artikulationspunkten hat also eine hohe praktische Relevanz.

Lern- und Aufklärungseffekt Die Aufgabenstellung und insbesondere das Erkennen der richtigen Antwort bewirken einen Lerneffekt. Damit hat die Aufgabe einen Wert, der über die Durchführung des Wettbewerbs hinausgeht. Die Beispielaufgabe verdeutlicht die Bedeutung des Aufbaus von Netzwerken. Es wird vermittelt, dass die Informatik sich mit der Ausfallsicherheit von Netzwerken beschäftigt. Konkret kann erkannt werden, dass sternförmige Netzwerkstrukturen kritisch sind.

Unterrichtliche Eignung Die Aufgabe eignet sich für die Verwendung im Unterricht. Gute Biber-Aufgaben können verwendet werden etwa als Impuls bei der Einführung eines Informatik-Themas, als Basisaufgaben für Formen des entdeckenden Lernens oder zur spielerischen Selbsteinschätzung der informatischen „Ader“. Die Beispielaufgabe kann beim Einstieg in das Thema Graphentheorie verwendet werden. Konzepte der Graphentheorie wie Zusammenhang, Zusammenhangskomponenten, Brücken und Artikulationspunkte können bei der Beschäftigung mit der Aufgabe entdeckt werden.

3.4 Inhaltskategorien

In einem Bebras-Wettbewerb sollen verschiedene Aspekte der Informatik ausgewogen vertreten sein. Um dies sicherstellen zu können, werden geeignete inhaltliche Kategorien

benötigt, denen die Aufgaben zugeordnet werden. Im Vorfeld des ersten Informatik-Biber in 2006 wurden bei Erstellung und Auswahl der Aufgaben folgende Kategorien entwickelt: (1) Logisches Denken, (2) Informationen verstehen, (3) Kombinatorik, (4) algorithmisches Problemlösen, (5) technische Kenntnisse, (6) allgemeines Wissen (z. B. über die Geschichte der Informatik). Diese Kategorien wurden als Vorschlag in die internationale Debatte eingebracht. Anfang 2007 wurden dann von Vertretern mehrerer Bebras-Länder, dem so genannten *Bebras Board*, folgende Kategorien entwickelt und zur Kategorisierung von Bebras-Aufgaben allgemein empfohlen (vgl. [DF08]):

ALG Der Kategorie *Algorithmic Thinking* (dt.: Algorithmisches Denken) werden alle Aufgaben mit algorithmischen Aspekten zugeordnet, einschließlich Aufgaben mit Bezügen zur Programmierung.

INF Die Kategorie *Information Comprehension* (dt. etwa: Informationen verstehen und begreifen) umfasst Aufgaben zur (symbolischen, numerischen, visuellen, ...) Repräsentation von Daten und Informationen, aber auch die Themen Kodierung und Verschlüsselung.

PUZ Logikrätsel und Spiele sind die Themen von Aufgaben der Kategorie *Puzzles* (dt.: Rätsel).

SOC Aufgaben der Kategorie *ICT and Society* (dt.: IKT und Gesellschaft) befassen sich mit sozialen, ethischen, kulturellen, internationalen und rechtlichen Aspekten.

STRUC Aufgaben zu diskreten Strukturen (z.B. Graphen) und aus dem Bereich Kombinatorik werden in die Kategorie *Structures, Patterns, and Arrangements* (dt.: Strukturen, Muster, Anordnungen) eingeordnet.

USE In die Kategorie *Using Computer Systems* (dt.: Computernutzung) fallen Aufgaben zum Einsatz von Informatiksystemen wie Standardanwendungsprogrammen, E-Mail(-Programmen), Suchmaschinen, usw. Aufgaben sollen sich auf allgemeine Prinzipien beziehen, nicht auf konkrete Produkte.

Diese Kategorisierung ist nicht endgültig und wird auf internationaler Ebene weiter diskutiert und entwickelt. Dabei werden auch mit anderer Zielsetzung entwickelte Kategorien für Informatikinhalte berücksichtigt werden, wie etwa die Inhaltsbereiche der deutschen Bildungsstandards Informatik [Pu08].

3.5 Aufgaben im Informatik-Biber 2008

Im Informatik-Biber 2008 wurden in jeder Altersgruppe 18 Aufgaben gestellt. Insgesamt wurden 37 verschiedene Aufgaben eingesetzt, einige also in mehreren Altersgruppen. Tabelle 1 zeigt die Zuordnung der Aufgaben zu den verschiedenen Inhaltskategorien. Die Kategorie USE war gar nicht vertreten; Aufgaben mit Bezug zum Einsatz von Informatiksystemen hatten immer auch einen ethischen oder rechtlichen Aspekt und wurden deshalb

Tabelle 1: Inhaltskategorien im Informatik-Biber 2008. Für jede Altersgruppe ist angegeben, welche Kategorien in den verschiedenen Schwierigkeitsstufen (*l*: leicht, *m*: mittel, *s*: schwer) und insgesamt (*ges.*) mit wie vielen Aufgaben vertreten waren.

	Benjamin				Junior				Senior				alle
	<i>l</i>	<i>m</i>	<i>s</i>	<i>ges.</i>	<i>l</i>	<i>m</i>	<i>s</i>	<i>ges.</i>	<i>l</i>	<i>m</i>	<i>s</i>	<i>ges.</i>	
ALG		2		2	1	1	3	5	2	1	4	7	10
INF	1		3	4	1	1	1	3	1	3	1	5	8
PUZ		2		2		1		1	1			1	3
SOC	4		1	5	3			3	2			2	7
STRUC	1	2	2	5	1	3	2	6		2	1	3	9
USE													

der Kategorie SOC zugeordnet. In die Kategorie der Logikrätsel (PUZ) gehörten nur wenige Aufgaben, während die anderen Kategorien annähernd ausgewogen vertreten waren. Aufgaben der Kategorie SOC wurden überwiegend als leicht eingestuft.

4 Durchführung des Informatik-Biber

Der Informatik-Biber wird in Deutschland als Online-Quiz durchgeführt. Dazu wird ein in Zusammenarbeit mit den Ausrichtern des niederländischen Bebras-Wettbewerbs entwickeltes Wettbewerbssystem genutzt. Teilnehmerinnen und Teilnehmer am Informatik-Biber müssen vor der Teilnahme ins Wettbewerbssystem eingetragen werden. Dies geschieht durch Lehrkräfte, die sich vorab als „Koordinatoren“ im System registrieren. Die Registrierungen werden von der BWINF-Geschäftsstelle geprüft. Die Koordinatoren legen die Zugangsdaten für das Wettbewerbssystem fest, mit denen sich die Jugendlichen im Wettbewerbszeitraum im System anmelden können. Nach Auswahl der richtigen Altersgruppe und Aufruf der Aufgaben läuft die Bearbeitungszeit; nach deren Ablauf beendet das System die Teilnahme automatisch. Der Wettbewerbszeitraum betrug bei den bisherigen Ausrichtungen des Informatik-Biber jeweils eine Arbeitswoche (Montag bis Freitag).

Auf Anregung von Koordinatoren wurde beim Informatik-Biber 2008 die Möglichkeit zur Teilnahme im Zwei-Personen-Team eingerichtet. So konnten je zwei Schüler gemeinsam einen Rechnerarbeitsplatz nutzen, was in vielen Schulen Standard bei der Nutzung von Computerräumen ist. In 2008 waren 12.442 der insgesamt 53.602 Teilnehmenden Mitglieder von Zweier-Teams.

Die starke Beteiligung am Informatik-Biber stellt in verschiedener Hinsicht eine Herausforderung dar, sowohl für die Ausrichter als auch für die beteiligten Schulen. Auf Ausrichterseite sind Registrierungs- und Anmeldeprozesse möglichst zu automatisieren, bei der Durchführung des Wettbewerbs ist die Verfügbarkeit des Wettbewerbssystems sicherzustellen. Auf der Seite der Schulen kann eine intensive Beteiligung die Grenzen der tech-

nischen Infrastruktur aufzeigen und stellt, etwa bei der Zuteilung der Arbeitsplätze, eine logistische Herausforderung dar.

Der lange Wettbewerbszeitraum (eine Arbeitswoche) erlaubt einerseits Teilnehmenden, Informationen über Aufgaben an spätere Teilnehmer zu übermitteln, andererseits aber auch eine breitere Teilnahme. Bei diesem Trade-Off zwischen möglichst breiter Teilnahme und sicherer Leistungsüberprüfung hat zur Zeit die breite Teilnahme Vorrang.

4.1 Teilnahme am Informatik-Biber

Seit der ersten deutschen Bebras-Ausrichtung mit 2.186 Teilnahmen aus 50 Schulen ist die Beteiligung an diesem Wettbewerb stark angestiegen. In 2007 wurde der Wettbewerb zum ersten Mal unter dem Namen „Informatik-Biber“ durchgeführt. Insbesondere bei Informatik-Lehrkräften wurde intensiv für den Wettbewerb geworben; es beteiligten sich 286 Schulen mit 21.802 Jugendlichen. Bei der letzten Ausrichtung im November 2008 waren 53.602 Jugendliche aus 417 Schulen dabei.

Tabelle 2: Teilnahme am Informatik-Biber 2008 nach Bundesländern; angegeben ist jeweils die Anzahl der Teilnehmenden und der Anteil an der Gesamtteilnahmezahl. Der Vergleichswert ist der Anteil an der Anzahl der Schülerinnen und Schüler aller Schulformen (also auch Primarstufe) im Schuljahr 2004/2005 in Deutschland (Quelle: Statistisches Bundesamt).

Bundesland	Anzahl Teilnehmende	Anteil	Vergleichswert
Baden-Württemberg	3.018	5,63%	13,73%
Bayern	12.465	23,25%	15,31%
Berlin	1.865	3,48%	3,70%
Brandenburg	3.241	6,05%	2,67%
Bremen	0	0,00%	0,77%
Hamburg	1.052	1,96%	1,89%
Hessen	1.264	2,36%	7,35%
Mecklenburg-Vorpommern	4.038	7,53%	1,77%
Niedersachsen	3.610	6,73%	10,32%
Nordrhein-Westfalen	10.407	19,42%	24,24%
Rheinland-Pfalz	2.493	4,65%	5,11%
Saarland	923	1,72%	1,22%
Sachsen	6.350	11,85%	3,80%
Sachsen-Anhalt	1.312	2,45%	2,40%
Schleswig-Holstein	145	0,27%	3,56%
Thüringen	1.419	2,65%	2,16%

Tabelle 2 enthält die Teilnahmewerte der einzelnen Bundesländer für den Informatik-Biber 2008. Das Verhältnis zwischen Teilnahmeanteil und Vergleichswert zeigt, wie die Teilnahmezahl eines Bundeslandes relativ zur Gesamtschülerzahl zu werten ist. Auf dieser Grundlage weist Mecklenburg-Vorpommern die relativ stärkste Teilnahme auf. Es fol-

gen Sachsen, Brandenburg und Bayern. Die starke Präsenz von Ländern mit Informatik-Pflichtunterricht in der Sekundarstufe 1 (wie etwa Bayern und Sachsen, vgl. [We07]) ist auffallend. Aber es gibt auch Beispiele für Bundesländer ohne Informatik-Pflichtstunden mit überproportionaler Beteiligung, insbesondere Brandenburg, aber auch das Saarland, Sachsen-Anhalt oder Thüringen. Nordrhein-Westfalen ist annähernd proportional beteiligt und hat als großes Bundesland absolut die zweithöchste Teilnehmezahl.

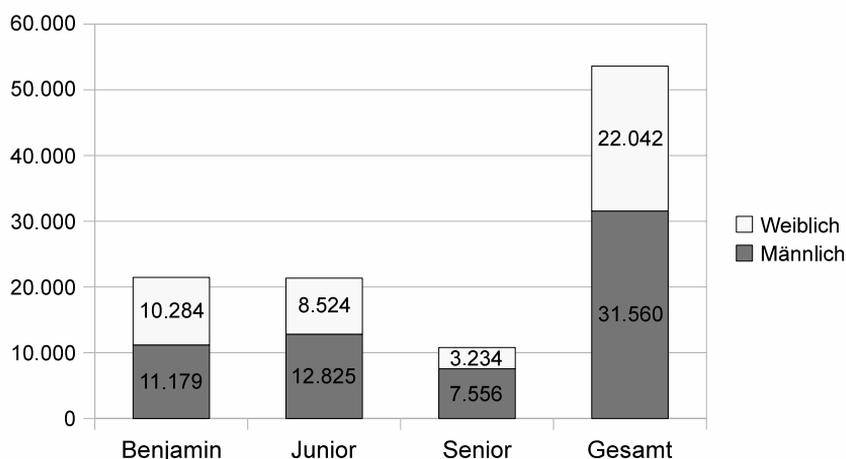


Abbildung 2: Anzahl der Teilnehmenden am Informatik-Biber 2008 nach Altersgruppe und Geschlecht. Die Mädchenanteile in Prozent lauten 48% (Altersgruppe Benjamin), 40% (Junior), 30% (Senior) und 41% (Gesamt).

In Abbildung 2 ist die Teilnahme am Informatik-Biber 2008 nach Altersgruppen und Geschlecht aufgeschlüsselt. Die absolut schwächere Teilnahme in der höchsten Altersgruppe geht grundsätzlich konform mit der geringeren Schülerzahl in der Sekundarstufe 2. Auffallend ist das Nachlassen der weiblichen Beteiligung mit zunehmendem Alter.

4.2 Internationale Entwicklung im Vergleich

Seit dem ersten Bebras-Wettbewerb in Litauen 2004 nahm die Zahl der beteiligten Länder stetig zu. In 2006 wurden Bebras-Wettbewerbe in Deutschland, Estland, Litauen, den Niederlanden und Polen veranstaltet. In Lettland, Österreich und der Slowakei wurden im Jahr 2007 erstmals Bebras-Wettbewerbe durchgeführt, in 2008 dann auch in Tschechien und der Ukraine. Insgesamt beteiligten sich in 2008 zehn Länder mit einem Wettbewerb an der Bebras-Initiative. Tabelle 3 gibt absolute und relative Teilnehmezahlen für die einzelnen Länder an. Deutschland weist die absolut stärkste Teilnehmezahl auf; relativ zur Bevölkerungszahl wurden die Bebras-Wettbewerbe von Estland, Litauen und Slowakei am besten angenommen. In diesem Vergleich ist allerdings nicht berücksichtigt, dass die An-

Tabelle 3: Teilnahme an den Bebras-Wettbewerben 2008. In der rechten Spalte ist die Anzahl der Teilnehmenden pro Tausend Einwohner angegeben. Die zu Grunde liegenden Einwohnerzahlen entstammen Angaben des Auswärtigen Amtes, Stand: Oktober 2008.

Land	Teilnahme absolut	Teilnahme relativ
Deutschland	53.602	0,65
Estland	4.039	3,01
Litauen	6.616	1,96
Lettland	700	0,31
Niederlande	5.120	0,31
Österreich	4.000	0,49
Polen	8.725	0,29
Slowakei	9.317	1,73
Tschechien	4.069	0,39
Ukraine	1.429	0,03
Gesamt	97.617	

zahlen der am jeweiligen Bebras-Wettbewerb beteiligten Jahrgangsstufen unterschiedlich sein können.

5 Bilanz der ersten Jahre

Mit beachtlichen und stark anwachsenden Teilnehmerzahlen hat der Informatik-Biber sich in der Landschaft der deutschen Schülerwettbewerbe etabliert. Die gesteckten Ziele konnten in weiten Teilen bereits erreicht werden:

Begegnung mit Informatik Die Teilnahmezahlen belegen, dass der Informatik-Biber vielen Jugendlichen und insbesondere solchen ohne Informatik-Vorkenntnisse eine (erste) Begegnung mit Informatik ermöglicht. Auch inhaltliche Vielfalt wurde erreicht: In den Aufgaben werden theoretische Grundlagen (z.B. Automaten, formale Sprachen, Graphentheorie) ebenso angesprochen wie praktische Aspekte (etwa Programmierung), informatische Denk- und Arbeitsweisen sowie Fragen des Einsatzes von Informatiksystemen (hier vor allem die Themen Sicherheit und Privatsphäre). Die international gesetzten Aufgabenkategorien sind im wesentlichen ausgewogen vertreten. Hierzu die Anmerkung einer Lehrkraft: „Ich habe schon von einigen Schülern aus den unteren Klassen gehört, dass sie nun ein viel klareres Bild von der Informatik haben als zuvor. Die Schüler bekommen ein sehr gutes Gefühl dafür, wie vielschichtig die Informatik doch tatsächlich ist.“

Werbung für Informatik Es hat sich gezeigt, dass auch dort, wo Unterricht in Informatik in nur geringem Maße vorhanden oder möglich ist, eine Teilnahme am Informatik-Biber realisierbar ist. Schulen, aber auch einzelne Lehrkräfte bekommen mit dem Wett-

bewerb die Möglichkeit, Informatik unabhängig vom Fachunterricht sichtbar zu machen, und zwar für Schüler, Eltern, Kollegium und Öffentlichkeit. Dazu trägt bei, dass nicht nur Teilnehmende, sondern auch besonders aktive Schulen mit Urkunden und Preisen ausgezeichnet werden.

Frühe Ansprache Mit dem Informatik-Biber werden Jugendliche bereits ab der 5. Klasse in großer Zahl auf Informatik aufmerksam. Das Auftreten des Wettbewerbs als Quiz mit eng begrenztem Teilnahmeaufwand kommt insbesondere jüngeren Schülern entgegen und macht die Teilnahme attraktiv. Besonders positiv ist der Mädchenanteil von etwa 40% im Vergleich mit anderen Wettbewerben zu werten. Am 27. Bundeswettbewerb Informatik z. B. nahmen etwa 10% Mädchen teil, in der vergleichbaren Biber-Altersgruppe Senior betrug der Mädchenanteil hingegen 30%. Der hohe Mädchenanteil wird dadurch begünstigt, dass Biber-Teilnahmen häufig kollektiv organisiert werden, also mit einer ganzen Klasse oder einem ganzen Kurs.

Heranführung an Informatik-Wettbewerbe Seit Einführung des Informatik-Biber steigt die Zahl der Teilnahmen am Bundeswettbewerb Informatik deutlich an: Die BWINF-Teilnahmezahlen in der ersten Runde der Jahre 2006 bis 2008 lauten 722, 868 bzw. 1106. Ein kausaler Zusammenhang mit dem Informatik-Biber lässt sich nicht nachweisen. Beim BWINF sind aber verstärkt Teilnahmen aus Schulen zu beobachten, die auch im Informatik-Biber aktiv sind. Einige dieser Schulen haben früher gar keine oder nur gelegentlich BWINF-Teilnehmer gestellt. Die Lücke bei der Ansprache von Jüngeren und Einsteigern kann der Biber nur teilweise schließen; der Abstand zwischen Biber und BWINF bzgl. des Anforderungsniveaus ist zu hoch.

Informatik-Begabung identifizieren Auf individueller Ebene zeigen Rückmeldungen, dass Schüler auf ihre Fähigkeiten im Bereich Informatik aufmerksam werden und zur weiteren Beschäftigung mit den so eingeführten Themenbereichen angeregt werden. Auf übergeordneter Ebene liefern die Wettbewerbsdaten mit den anonymisierten Bearbeitungsergebnissen der teilnehmenden Schüler reiches Material für querschnittliche und langfristige auch längsschnittliche Statistiken wie den Vergleich der Teilnahme- und Erfolgsquote nach Altersgruppe, Geschlecht, Schulart oder Bundesland. Verknüpft mit einer Klassifizierung der Aufgaben, etwa der inhaltlichen Kategorisierung (Abschnitt 3.4) und der Schwierigkeitsstufung (Abschnitt 3.1) werden bereits Analysen zu spezifischen Kompetenzmerkmalen durchgeführt [Sc08].

Internationale fachdidaktische Kooperation Die gemeinsame Diskussion des Aufgabenpools durch alle beteiligten Länder und der derzeit gehaltene Konsens bzgl. der Inhaltskategorien für Aufgaben haben dazu geführt, dass sich die Vorstellungen der einzelnen Länder von einem guten Bebras-Wettbewerb schon deutlich einander angenähert haben. Dieser Effekt wird sich verstärken, wenn immer mehr gemeinsame Aufgaben gefunden werden müssen. Es gibt aber immer noch starke Unterschiede, sowohl inhaltlicher als auch organisatorischer Natur.

6 Ausblick

Der große quantitative Erfolg des Informatik-Biber und die sehr positive Resonanz auf die Wettbewerbsinhalte bestärken die Verantwortlichen, die Entwicklung des Wettbewerbs weiter voranzutreiben. Die bisher (insbesondere wegen geringer Ressourcen) überwiegend ad-hoc betriebene Aufgabenentwicklung soll professionalisiert und die in den Abschnitten 3.2 und 3.3 genannten Anforderungen sollen noch stärker durchgesetzt werden. Das Wettbewerbssystem ist weiterzuentwickeln, um den steigenden Teilnehmezahlen gerecht zu werden und interaktiv zu bearbeitende Aufgaben zu ermöglichen. Die Bebras-Wettbewerbe der einzelnen Länder sollen einander noch näher gebracht werden, in inhaltlicher wie organisatorischer Hinsicht. Im Informatik-Biber sind Maßnahmen zu treffen, wie eine Leistungsverfälschung trotz der lockeren Teilnahmebedingungen möglichst verhindert werden kann. Schließlich soll die Teilnahme und insbesondere die Zahl der beteiligten Schulen weiter gesteigert werden.

Danksagung

Im Aufgabenausschuss des Informatik-Biber 2008 waren aktiv (in alphabetischer Reihenfolge) Hans-Werner Hein, Wolfgang Pohl, Kirsten Schlüter, Renate Thies, Marco Thomas und Michael Weigend. Für Unterstützung bei der Umsetzung des Wettbewerbs bedanken wir uns bei Miriam Bastisch, Paul Hooijenga und Eljakim Schrijvers.

Literatur

- [Da06] Valentina Dagiène. Information Technology Contests – Introduction to Computer Science in an Attractive Way. *Informatics in Education*, 5(1):37–46, 2006.
- [DF08] Valentina Dagiène und Gerald Futschek. Bebras International Contest on Informatics and Computer Literacy: Criteria for Good Tasks. In Roland Mittermeir und Maciej Syslo, Hrsg., *Informatics Education – Supporting Computational Thinking*, LNCS 5090, Seiten 19–30, Berlin Heidelberg, 2008. Springer-Verlag.
- [Pu08] Hermann Puhlmann et al. Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. Beilage zu LOG IN, Heft 150/151, 2008.
- [PKH07] Wolfgang Pohl, Katharina Kranzdorf und Hans-Werner Hein. Einstieg Informatik - Aktivitäten und Erfahrungen. In Sigrid Schubert, Hrsg., *Didaktik der Informatik in Theorie und Praxis*, Lecture Notes in Informatics, Seiten 253–264, Bonn, 2007. Gesellschaft für Informatik.
- [Sc08] Kirsten Schlüter. Je schwieriger die Aufgabe, desto klüger der Kopf? In Torsten Brinda, Michael Fothe, Peter Hubwieser und Kirsten Schlüter, Hrsg., *Didaktik der Informatik – Aktuelle Forschungsergebnisse*, Seiten 77–86, Bonn, 2008. Köllen.
- [We07] Moritz Weeger. Synopse zum Informatikunterricht in Deutschland, Stand: 16.11.2007. <http://dil.inf.tu-dresden.de/schule/Weeger/output.inf.tu-dresden.de/homepages/index9c3a.html>, 2007.

Informatik-Schnuppereinheiten zur Positionierung des Unterrichtsfachs

Ernestine Bischof, Roland T. Mittermeir

Institut für Informatik-Systeme
Universität Klagenfurt
Österreich
{ernestine, roland}@isys.uni-klu.ac.at

Abstract. Informatikunterricht im allgemeinbildenden Unterricht verfehlt heute sehr oft das Ziel, Begeisterung für dieses Fach zu wecken. Im Gegenteil, zu starke Fokussierung auf Anwendungsschulung vermittelt ein falsches Bild von Informatik und führt dazu, dass viele an sich Qualifizierte, vor allem qualifizierte Mädchen, als künftige Informatikstudenten verloren gehen. Das hier vorgestellte Konzept aufbauender Schnuppereinheiten versucht, diesem Trend entgegenzuwirken. Ziel ist, Kinder und Jugendliche in altersgerechter Form an Thematiken des technischen Fachs *Informatik* heranzuführen und ihnen so zu zeigen, dass sich weitere Beschäftigung mit Fragestellungen der Informatik intellektuell lohnt.

1 Motivation

Als Informatik als Schulunterrichtsfach eingeführt wurde, war es für viele Jugendliche ein faszinierendes Fach. Es galt als modern und zukunftsgerichtet, wenngleich nicht immer leicht. Dieser Befund gilt, wie in persönlichen Gesprächen sowie in gezielten Interviews [AKLU07] und umfangreichen Befragungen [Mich08] feststellbar, heute nicht mehr. Der Grund dafür mag zu einem guten Teil darin liegen, dass Computer die Aura des Besonderen verloren haben und in Form des Personal Computers zu Gegenständen unseres täglichen Lebens geworden sind. Zu einem weit stärkeren Teil mag aber Form und Inhalt des Unterrichts, den Kinder und Jugendliche unter der Überschrift Informatik vermittelt bekommen, für diese Umkehr der Wertigkeit verantwortlich sein.

War Informatikunterricht in seinen Anfängen weitestgehend durch das Lösen von Programmieraufgaben geprägt [Reit05], so dominiert heute wenigstens an vielen österreichischen Schulen ein an den Ausbildungszielen der European Computer Driving Licence (ECDL) orientiertes Curriculum. Dieser Wechsel kann begründet werden; etwa dadurch, dass nicht eine ganze Population Jugendlicher zu Software-Entwickler/innen ausgebildet werden soll. Auch gibt die Orientierung am ECDL Lehrkräften, die sich mit einem sehr offenen Lehrplan konfrontiert sehen, eine Handreichung bei der Stoffauswahl. Darüber hinaus erfolgt durch die schulexterne ECDL-Prüfung sowohl eine Zertifizierung des Unterrichts als auch der Unterrichteten.

Allerdings ging dadurch die intellektuelle Faszination am Fach, wie sie etwa von Dagsys, Dagiene und Grigas in der Beschreibung der Anfangsphasen des Informatikunterrichts, in der Programmieraufgaben in Zeitungen publiziert wurden und jene, die sie lösten, ihre Programme zur Überprüfung an ein Universitätsinstitut senden konnten [DDG06], verloren. In einer von Micheuz [Mich09] an österreichischen Schulen durchgeführten Befragung umschrieben Jugendliche der 9. Schulstufe die Inhalte des Informatikunterrichts dominant mit *Word*, *Textverarbeitung*, *Excel*, *langweilig*, *Präsentation*. Begriffe wie *spannend* oder *interessant* folgten erst mit deutlichem Abstand. Dabei fällt nicht nur auf, dass Produktnamen vor dem generischen Begriff kommen, sondern vor allem, dass der Unterricht des Texteschreibens offenbar so dominant ist, dass er sogar beide Spitzenpositionen einnimmt. In diesem Kontext sollte jedoch der hohe Rang von „langweilig“ oder „uninteressant“ nicht sehr wundern. Mögen sich doch all jene, die Maschinschreibunterricht besuchten, an diesen zurückerinnern. War dieser intellektuell anspruchsvoll? Hätte er getaugt, für ein anschließendes Maschinschreibstudium zu motivieren?

Wir haben mithin erkannt, dass der an vielen österreichischen Schulen mit bester Absicht praktizierte Informatikunterricht kaum geeignet ist, die kreativsten Köpfe für ein nachfolgendes Informatikstudium zu motivieren. Insbesondere aus einer an zwei Schulen durchgeführten Studie [AKLU07] ging hervor, dass speziell bei Mädchen negative Vorurteile gegenüber Informatik und Informatikfreaks bestehen, die zwar einem Zerrbild entsprechen, jedoch durchaus ernst zu nehmen sind.

Ausgehend von diesen Überlegungen starteten wir ein Projekt, das Fach Informatik im Bewusstsein Jugendlicher dort zu positionieren, wo es sich aus der Eigensicht tatsächlich befindet. Wir wurden dabei durch einen Beirat von Schulpädagog/innen, die in unterschiedlichen Schulstufen tätig sind, begleitet. Insbesondere sollte klar gestellt werden, dass Informatik ein konstruktives technisches Fach ist. Damit sollte die Verwechslung zwischen Fahrschule und Konstruktionsbüro aus der Welt geschaffen werden. Allerdings ist dabei zu berücksichtigen, dass der Unterschied zwischen Fahrschule und Konstruktionsbüro leichter nachvollziehbar ist, als in Informatik. Die unterschiedlichen Fahrzeugtypen und Modelle sieht man ja, man kann sie auch ausprobieren und erlebt so wenigstens die Effekte der Tätigkeiten der Konstrukteur/innen. Um Informatik erlebbar zu machen, bedarf es einiger Umwege, die im folgenden Kapitel dargestellt werden.

2 Informatik erLeben – Konzept einer Zurechtrückung

Ausgehend von obigem Befund versuchten wir, ein didaktisches Konzept zu entwickeln, das in Analogie zu Fahrschule – Konstruktionsbüro – (Fertigungsstraße) – Autofahren für den Informatikbereich die Unterschiede zwischen ECDL – Software-Konzeption – (Software-Fabrik) – Software-Anwendung verdeutlicht. Weiters sollte in Burschen wie vor allem auch in Mädchen Interesse am späteren Studium eines technischen Fachs wie Informatik geweckt werden.

Allerdings sind zum Unterschied von Maschinenbau oder Bauingenieurwesen/Architektur die Produkte von Informatikern nicht (Software) oder nicht in ihrem eigentlichen Wesen (Hardware) sichtbar und daher in ihrer Funktionsweise nicht erlebbar. Hilfestellung bei der Überwindung dieser Hürde bot Antonitschs Idee, Kara-Programme [RNH04] auf den Fliesen des Schulgangs durch Schüler/innen nachspielen zu lassen [ALS07]. Diese Animation birgt eine Fülle von Implikationen:

- Das Konzept des Automaten als beschränktes, strikt Anweisungen befolgendes Gerät wird durch die Kara simulierende Person deutlich. Es wird insbesondere dann deutlich, wenn diese Person versucht, intelligenter zu sein, als die eben erhaltene Anweisung und die Beobachter der Szene auf das Missverhältnis zwischen Programm und Handlung aufmerksam werden.
- Es wird klar, dass *Programmieren* keine magische Tätigkeit ist, sondern einfach darin besteht, exakt zu formulieren und künftige Situationen in den niedergeschriebenen Verfahrensanweisungen zu antizipieren.
- Ebenso wird klar, dass es, um die Prinzipien des *Programmierens* zu verstehen und zu erlernen, keiner besonderen, irgendwie kryptisch erscheinenden Programmiersprache bedarf. Wir müssen uns nur einer Sprache bedienen, die der Adressat unserer Anweisungen, sei es Mensch oder Maschine, versteht.
- Das Verfahren wird von der „programmgesteuerten Person“ so langsam ausgeführt, dass die gesamte Klasse den Vorgang im wahrsten Sinn des Wortes Schritt für Schritt nachvollziehen kann.
- Letztlich konnte festgestellt werden, dass diese Einheiten den Jugendlichen Spaß bereitet haben und das ist bei gutem Unterricht ja nicht verboten.

Aus obigen Punkten wird klar, dass derartige Interventionen in den Informatikunterricht zweifellos allgemeinbildenden Charakter haben. Schärfung des Ausdrucks, Berücksichtigung der sprachlichen (und gegebenenfalls auch intellektuellen) Fähigkeiten des „Kommunikationspartners“, Antizipation von Situationen, etc., sind Fähigkeiten, die in der Informatik wie im Leben jedes Menschen wichtig sind. Sich darin zu üben hat also wenig mit künftigen Informatikberufen zu tun. Es ist aber durchaus anspruchsvoll und ist auf einer anderen als auf der eher sportlichen Skala zu bewerten, die misst, wie viel Nettoanschläge pro Minute man in Maschinschreiben schafft oder wie viel Word- und/oder Excel-Befehle man auswendig kann. Simulation und Spiel stellen mithin eine Leitlinie für unser Projekt *Informatik erLeben* dar.

Eine weitere Rahmenbedingung wurde uns von den Schulpraktikern auferlegt. Sie meinten, man müsse mit einem derartigen Projekt auf Volksschulebene beginnen, da im Alter von acht bis zehn Jahren noch ausreichend Begeisterungsfähigkeit vorläge und somit wesentliche Weichen für den späteren schulischen wie außerschulischen Bildungsweg gelegt würden. Dies schraubte die Hürde für unser Vorhaben noch etwas weiter in die Höhe, da dadurch der noch beschränkte Vorrat an mathematischem Vorwissen, und die noch beschränkte Abstraktionsfähigkeit der Zielgruppe berücksichtigt werden mussten.

Schlussendlich waren wir der Meinung, dass eine Intervention auf Volksschulebene wenig bringt, wenn nicht eine weitere, wenn auch nur punktuelle, Betreuung über die restliche Schulzeit aufrecht erhalten werden könnte.

Auf dieser Grundlage erarbeiteten wir einige Themenstränge, in denen Informatikinhalte (oder auch allgemeinere Technikinhalte) so dargestellt werden konnten, dass sie in altersgerechten Schritten Technik- und Informatikverständnis vermitteln. Der Beirat wählte dabei insbesondere jene Themen aus, die für die oberen Grundschulklassen (in Österreich 3. und 4. Klasse Volksschule) geeignet erschienen. Da sich einige dieser Themen nicht für eine durchgängige Fortsetzung bis zu Abiturklassen anbieten (z.B. Bilder und Farben) oder am Weg dorthin Lücken aufwiesen (z.B. Verschlüsselung), ergänzten wir die Themen noch um einige Gebiete, die uns fachlich besonders am Herzen lagen. Aus diesen Überlegungen resultierten die in Kapitel 4 beschriebenen neun Themenstränge.

Die für die Intervention in Schulen vorgesehenen Einheiten umfassen jeweils zwei Schulstunden (oder 100 Minuten). Die meisten Einheiten sind weiter gegliedert in Module. Dadurch können auch themenstrangübergreifende Topoi besprochen werden. Einige didaktische Grundüberlegungen sollten dabei beachtet werden:

- Insbesondere für die noch sehr junge Zielgruppe sollte wenigstens einer der Module stark spielerischen Charakter haben. Aber auch ältere Jugendliche sollten etwas mitnehmen können, mit dem sie in einer Peer-Gruppe beeindrucken können (Etwa: „Wie weit kannst Du mit den Fingern beider Hände zählen?“ – „Ich schaffe es bis über 1000!“)
- Wo immer möglich sollten Verbindungen zur Erfahrungswelt der Kinder und Jugendlichen aufgebaut werden.
- Animationsanteile der Module sollten zur Beobachtung anregen. Schulung des Beobachtungsvermögens sollte Beobachtungswillen und Neugierde stimulieren.
- Die Einheiten enthalten viele Fragen an die Klasse. Weitere Fragen sollten angeregt werden.
- Dass Informatik (und Technik allgemein) auf naturwissenschaftlichen Grundlagen wie Mathematik und Physik beruht, wird nicht verschwiegen. Im Gegenteil! Aber diese Grundlagen müssen so einfach gebracht werden, dass sie für die Kinder bzw. Jugendlichen der jeweiligen Altersgruppe nachvollziehbar sind.
- Dabei, und natürlich auch bei den Informatikinhalten im engeren Sinn, darf allerdings nie auf Korrektheit verzichtet werden. Diese schwierige Randbedingung muss durch Wahl des richtigen Abstraktionsniveaus eingehalten werden. Analogien wirken unterstützend, dürfen aber nie überzogen werden. Das Feld, aus dem die Analogien stammen, ist weit. Es reicht von Kunst bis zur Natur.
- Unter den Fragen sollten auch solche sein, die schon einiges an Mit- und Nachdenken verlangen. Aber wir wollen sicherlich keinen Prüfungsstoff vermitteln.
- Wenn dies möglich ist, werden kontrastierende Verfahren vorgestellt, um zu zeigen, dass sich Nachdenken lohnt und dass Informatik kein naives Bastel- und Probierfach ist, auch wenn das Experiment einen wichtigen Stellenwert hat.
- Auch sollte klar gestellt werden, dass Informatikerinnen und Informatiker nicht die Maschine zum Ziel ihrer Tätigkeit haben, sondern dass die Maschine vielmehr ein Instrument ist, das im Rahmen eines Gesamtkonzepts in einer für andere Menschen sinnvollen und effizienten Weise eingesetzt wird.

Aus diesen Randbedingungen folgt, dass hier kein abgeschlossenes, umfassendes didaktisches Konzept realisierbar ist, sondern nur lose aufeinander aufbauende, aber in sich abgeschlossene Interventionseinheiten angeboten werden.

4 Verfügbare Einheiten

Die Umsetzung dieser Ziele erfolgt in neun Themensträngen, die in einzelne Einheiten bzw. Module gegliedert sind. An einigen Stellen bestehen explizite Querbeziehungen, sodass auch Einheiten aus verwandten Modulen unterschiedlicher Stränge (etwa Binärsystem und Addierwerk) zusammengestellt werden können.

4.1 Bilder, Grafik und Zeichnen

Dies mag nicht als zentrales Thema der Informatik erscheinen. Da die ins Auge gefassten Zielgruppen aber schon im Grundschulbereich beginnen, bietet es einen Anknüpfungspunkt zum Vorwissen der Kinder und kann gut spielerisch aufgearbeitet werden. Im Modul B1 werden physikalische Grundlagen zur Farbwahrnehmung be-



Abbildung 1: SchülerInnen der Primarstufe bei einer Informatik-erLeben Piloteinheit

handelt. Darauf aufbauend können die Kinder im Modul B2 zur Farbsynthese selbst die Druckfarben Gelb, Magenta und Cyan mischen. Sie lernen so die Unterschiede zwischen additiver und subtraktiver Farbmischung kennen. Ebenfalls bereits ab der Primarstufe empfohlen ist Modul B3 Grafikformate, der erklärt wie Bilder im Computer gespeichert werden können, obwohl dieser nur auf der Basis von 0 und 1 arbeiten kann. Ein weiteres Grundschulmodul ist die Druckausgabe.

Das Thema Farbtiefen ist für Klassen ab der Sekundarstufe I vorgesehen. Abb. 1 zeigt Schüler/innen einer 3. Volksschulklasse beim Ausmalen von Farbkreisen mit den Druckerfarben. Für uns war diese Einheit wichtig, da sich zeigte, dass aufgrund der noch eingeschränkten Motorik die Vorlagen größere Malflächen erfordern.

4.2 Codierung

Mit der Codierung nähern wir uns bereits den Kernthemen der Informatik. In den Einheiten zur Codierung werden wichtige Verständnisgrundlagen für die Informatik geschaffen. Im Grundschulbereich beginnen wir wieder spielerisch, etwa mit einem „Morsespiel“ in Modul C1. Anhand des Morsecodes wird das Grundprinzip der Zuordnung von Zeichen eines Alphabets zu Zeichen eines anderen Alphabets veran-



Abbildung 2: Fehlerkorrektur mit Prüfbits in der Hauptschule Kühnsdorf (Sekundarstufe I)

schaulich. Im darauf aufbauenden Modul können die Kinder selbst einen Code mit verschiedenen Farben entwickeln. Anschließend können sie dann ihre Namen im Morsecode und in einem selbst entwickelten Code codieren. In weiteren Modulen werden Codebäume eingeführt, in der Primarstufe noch sehr spielerisch, ab der Sekundarstufe sollen dann auch bereits eigene kleine Codierungen mit Codebäumen entstehen. Codeoptimierung und Huffman Codierung werden ebenso thematisiert. Das Modul C4 bietet ab der Sekun-

darstufe I eine Einführung in das Binärsystem basierend auf Ideen in [BWF06]. Fehlererkennung und Fehlerkorrektur werden, wie obiges Foto zeigt, im Modul C6 für die Sekundarstufe spielerisch thematisiert ([BWF06]).

4.3 Verschlüsselung

Mit einfachen Verschlüsselungsverfahren, wie der Cäsar-Verschlüsselung beginnen wir bereits in Grundschulklassen spielerisch einfache Sätze zu ver- und entschlüsseln. Ab der Sekundarstufe II kann man mit den Schüler/innen bereits verschiedene symmetrische Verschlüsselungsverfahren, wieder ausgehend von der Cäsar-Verschlüsselung, vergleichen und ihre Sicherheit abwägen. Da die Verschlüsselung bei der virtuellen Kommunikation immer wichtiger wird, kann hier den Kindern und Jugendlichen auch starker Praxisbezug vermittelt werden. In der Sekundarstufe II werden auch komplexere Verfahren, wie öffentlicher Schlüsseltausch und Public-Key-Verschlüsselungsverfahren (RSA) thematisiert.

4.4 Hardware

Der Computer wird in der Schule und von unterschiedlichen Berufsgruppen im Alltag verwendet. Was nun aber wirklich in einem Computer drinnen steckt und wie er rechnet, wissen nur wenige. In den Einheiten H1 bis H3 sollen die Schüler/innen altersgerecht die grundlegende Rechenweise des Computers kennen lernen. In Animationen übernehmen sie die Funktion der Computerbauteile. Im Grundschulbereich berechnen sie einfache Additionen oder Sub-



Abbildung 3: Volksschülerin mit Motherboard

traktionen, später komplexerer Rechenabfolgen in einer vereinfachten Maschinensprache. Um zusätzlich zum Verständnis auch noch ein Bild von der Hardware zu vermitteln, werden im Anschluss nicht mehr gebrauchte Rechner geöffnet und die einzelnen Bauteile benannt. Im Modul H4 stellen die Schüler/innen die Funktionsweise eines Halb- und Volladdierers dar.

Abb. 3 zeigt eine Schülerin einer 4. Volksschulklasse mit einem Motherboard. Vorher spielten die Kinder mit einfachen Rechenaufgaben die Funktionsweise einer CPU. Besonderes Interesse erweckte auch eine zerlegte Hard-Disk.

4.5 Netze

Der vielfältige Bereich der Computernetze wird für Grundschul Kinder spielerisch, etwa mit einem „stille Post“-Spiel aufgearbeitet. Wie es bei der Kommunikation zwischen Menschen Höflichkeitsformen und Kommunikationsregeln gibt, so gibt es beim Nachrichtenaustausch zwischen Computern auch Protokolle, die gewisse Regeln festlegen. Von diesen elementaren Grundlagen kommen wir dann in der Sekundarstufe zu den Netzwerkprotokollen und Netzwerktypen. Was passiert aber, wenn wir im Auto mit dem Mobiltelefon telefonieren oder im Internet surfen? Wie bleibt die Verbindung aufrecht? Diese Themenbereiche werden mit Schüler/innen der Sekundarstufe II besprochen.

4.6 Betriebssysteme

Vielen Anwendern ist der Unterschied zwischen Betriebssystem und Anwenderprogrammen nicht wirklich bewusst. Einige wissen, dass beide zur Software zählen, andere wissen nicht einmal, dass man zwischen den beiden unterscheidet. Die zentrale Rolle des Betriebssystems wird für Kinder der Primarstufe mit einer Apothekerin (Betriebssystem) verglichen, die Kunden (Prozesse, Anwenderprogramme) bedient. Sie verteilt die Ressourcen, im Computer sind dies etwa die CPU, Speicher oder der Drucker, an die einzelnen Anwendungsprogramme. Ab der Sekundarstufe gehen wir dann auch bereits auf Scheduling-Strategien bei der Vergabe dieser Ressourcen ein, sowie auch auf mögliche Probleme wie Deadlocks oder Livelocks.

4.7 Programmieren

Programmieren im Grundschulalter, und das auch noch ohne Computer? Wir beginnen hier im Modul P1 mit einfachen Wegbeschreibungen, etwa innerhalb des Schulgebäudes, welche die Kinder in freiem Text verfassen. Um auf die Bedeutung von Präzision in solchen Beschreibungen/Algorithmen aufmerksam zu machen, probieren die Kinder ihre Beschreibungen gegenseitig aus. So können Fehler oder Ungenauigkeiten im Algorithmus entdeckt werden. Soll ein Algorithmus für einen Computer ausführbar sein, muss die Beschreibung exakt und eindeutig sein. So sollen auch bereits kleinere Kinder zum Algorithmus-Begriff hingeführt werden, allerdings noch ohne zu programmieren. Im Modul P2 sollen die Wegbeschreibungen bereits mithilfe

einfacher Kontrollstrukturen wie Folgen, Alternativen (Wenn – Dann) oder Schleifen (Wiederholung) strukturiert werden. Die Module P3 und P4 behandeln unterschiedliche Sprach- und Kommunikationsstile. Diese Module können in höheren Klassen als Einstieg in die Programmierung bzw. in Objektorientierung verwendet werden. In Modul P5 wird das grundlegende Denkmuster der Rekursivität anhand einfacher Beispiele aus Kunst, Natur, Sprache und Mathematik vermittelt.

4.8 Sortieren

In den Einheiten und Modulen zu Sortieralgorithmen sollen die Jugendlichen Einblick bekommen, wie nützlich das Sortieren von Daten ist. Um aber überhaupt einmal einige Sortierverfahren kennen zu lernen, spielen die Schüler/innen das Verfahren in einer Animation durch. Sie stellen sich in einer ungeordneten Reihe auf und werden von einer weiteren Person dann etwa nach dem Geburtsdatum sortiert. Sie sollen dazu hingeführt werden, verschiedene Verfahren nach ihrer Geschwindigkeit bzw. Effizienz zu vergleichen. Neben Bubblesort (So1) und Selectionsort (So2), die auch bereits in der Grundschule ansatzweise vorgestellt werden können, werden auch komplexere Verfahren wie Mergesort (So3) und Quicksort (So4) vorgestellt und bewertet.

In einer Übungseinheit entwickelten die Teilnehmer selbständig ein an Radixsort angelehntes Verfahren. Dies zeigt, dass es sich bei all diesen Interventionen lohnt, mit Fragen zu beginnen und mit dem vorbereiteten Konzept recht offen umzugehen. Manchmal ergeben sich spontan Situationen, die exzellent in das didaktische Konzept passen und so die Motivation erhöhen. Freilich zeigt eben dieses Beispiel, dass die Systematisierung des Vorschlags nach Kategorien innerhalb von Rechnungsnummern zu sortieren für einige Teilnehmer/innen nicht mehr ganz nachvollziehbar war.

4.9 Suchen

Suchalgorithmen gehören, wie Sortieren, zu den Grundlagen der Informatik und stellen ein zentrales Thema dar. Suchstrategien hat bereits jeder von uns unabhängig von informatischen Strukturen entwickelt, sei es ob man in einem Telefonbuch sucht oder in ungeordneten Strukturen. Daher wird mit elementaren Suchstrategien bereits in der Grundschule begonnen, etwa im Modul Su1 mit der blinden Suche in einem verschlossenen Sack. Die Kinder sollen intuitiv eine Suchstrategie anwenden und diese dann für die anderen Kinder beschreiben. Auf den ansteigenden Schwierigkeitsgrad diese Aufgabe auszuführen und dann ihre Lösung anschließend zu verbalisieren wird in [Kol08] hingewiesen. Diese Übung eignet in Kombination mit den Modulen P1 oder P2 auch als Vorbereitung auf Programmierung.

Darauf aufbauend soll in den folgenden Modulen erkannt werden, wie der Suchaufwand von der Struktur der Daten abhängt, etwa ob wir eine unsortierte oder eine sortierte Struktur vor uns haben. Um Praxisbezug herzustellen, kann auf Suche nach kritischen Messdaten hingewiesen werden. Hier kann ein schneller Treffer lebensrettend sein.

5 Wege zum Ziel

Wir sehen zwei Wege vor, die eben beschriebenen Interventionseinheiten an Schülerinnen und Schüler heranzutragen.

Der erste Weg ist, dass wir anbieten, selbst oder durch Kolleginnen, die die Entwicklung der Einheiten verfolgten und wohlmeinend kritisierten, Schulbesuche abzustatten, in denen wir mit der Klasse eine zweistündige Unterrichtssequenz gestalten. Dies erfolgt nach vorhergehender Rücksprache mit der Informatiklehrkraft oder der Klassenlehrkraft (in Volksschulen gibt es in Österreich keinen gesondert ausgewiesenen Informatikunterricht, aber im Sachunterricht kann auf Informatik Bezug genommen werden). Diese Lehrkraft kennt die Klasse und bestimmt daher das Thema der konkreten Intervention.

Um Lehrkräfte auf diese Möglichkeit aufmerksam zu machen stellten wir diese Option in dem Heftchen *Informatik erLeben* [BiMi08] vor. Es konnte Dank einer Förderung des Kärntner Wirtschaftsförderungsfonds und des IT-Campus allen einschlägigen Lehrerinnen und Lehrern Kärntens zur Verfügung gestellt werden. Dieses Heftchen sollte jedoch aufgrund einer dringenden Empfehlung des Beirats kurz gehalten sein. Es enthält daher neben einer Vorstellung des Projekts lediglich 5 Module (zwei zu Codierung, einer zu Verschlüsselung und zwei kontrastierende Sortierverfahren). Die übrigen Einheiten werden schrittweise auf der Web-Site <http://informatik-erleben.uni-klu.ac.at> publiziert.

Der zweite Weg führt teils über den ersten, teils wird er direkt angestrebt. Eine Grundidee des eben beschriebenen Wegs ist, dass Lehrkräfte unser Wirken in der Klasse beobachten und künftig die Einheiten, die wir mit den Kindern bzw. Jugendlichen übten, selbst durchführen. Dieser Weg kann natürlich auch direkt besritten werden. Die von uns verwendeten Materialien sind in der Regel billig, teils sind sie über die oben angegebene Web-Site beziehbar, teils werden sie, wie etwa ein veraltetes Computer, den man zerlegen und wieder zusammenbauen kann, an der Schule ohnehin vorhanden sein.

Dieser zweite Weg bereitete uns allerdings etwas mehr Arbeit als ursprünglich geplant. Die Ausarbeitungen der Module und Einheiten waren ursprünglich so, dass wir und wohl auch jede/r andere Absolvent/in eines Lehramtsstudiums Informatik anhand dieser Skizzen den vorgezeichneten Unterricht abhalten könnte. Hier sprang der Beirat allerdings wieder warnend ein. Nur wenig Informatik Unterrichtende sind Absolventen einer einschlägigen Studienrichtung. Daher sind die Materialien zwar als Lehrbehelfe zu gestalten, doch dies hat so zu erfolgen, dass sie gleichzeitig auch selbst die Rolle von Lehrmaterialien für Lehrende übernehmen können.

Zum Glück hilft uns dabei die Technologie, die es zu vermitteln gilt. Da wir viele der Einheiten schon mit Kindern erprobt haben, konnten wir uns über deren Angemessenheit Klarheit verschaffen. Für die bereitgestellten Materialien können wir dies noch nicht behaupten. Da sie jedoch via Web-Site publiziert sind, muss man bei

allfälligem Ergänzungsbedarf nicht auf eine Neuauflage der Sammlung warten. Nötige Aktualisierungen oder Korrekturen können fallbezogen erfolgen.

6 Erfahrungen bei bisherigen Schulbesuchen

6.1 Eindruck aus den Schulen

Viele der vorgestellten Einheiten wurden im Sommersemester 2008 bereits in einer Pilotphase an Schulen durchgeführt. An der Pilotphase waren 3 Grundschulklassen, 3 Klassen aus der Sekundarstufe I, sowie 3 Klassen der Sekundarstufe II beteiligt. Es zeigte sich grundsätzlich positives Feedback und großes Interesse seitens der Schulen. Diese Phase diente allerdings auch zur Verbesserung der Einheiten und zum Einholen konstruktiver Kritik. So zeigte sich etwa, dass bei Einheiten im Sekundarstufenbereich teilweise kleine inhaltliche Vorbereitungen durch den Klassenlehrer nötig sind. Besonders im Grundschulbereich merkten wir auch, dass die Haltung der Klassenlehrerin sowie auch die Motivation der Kinder durch die Klassenlehrerin Einfluss auf das Gelingen oder Misslingen einer Informatik-erLeben-Einheit haben.

Weiters war es sehr wichtig, nicht starr am Konzept der Einheit festzuhalten, sondern bei Bedarf spontan zu reagieren, wenn man merkt, dass die Kinder oder Jugendlichen nicht mehr aufnahmefähig sind. So war es einmal nötig die Einheit abubrechen und zur Auflockerung ein neues Thema zu beginnen. Grundsätzlich hat sich gezeigt, dass Kinder vor allem bis zur Mitte der Sekundarstufe I sehr begeisterungsfähig sind. Aktive Beteiligung ist allerdings auch in Einheiten für höhere Schulstufen nötig.

Überraschend und erfreulich waren Erfahrungen, die das Interesse der Schüler/innen zeigten. So begann beispielsweise ein Schüler der 5. Schulstufe während der Verschlüsselungseinheit, selbst eine Verschlüsselungsart zu entwerfen, indem er Buchstaben durch Zeichen ersetzte.

5.2 Schüler/innenrückmeldungen

Um auch das Feedback der Schüler/innen einzuholen, wurde in den Grundschulklassen und in einer Klasse der Sekundarstufe I ein Fragebogen ausgegeben. Von den insgesamt 59 Schüler/innen der Grundschulklassen gaben 53 an, dass sie das vorgestellte Thema interessiert hat. 47 gaben an, dass sie sich im Allgemeinen für Technik interessieren. Auf die Frage, ob die *Informatik-erLeben*-Einheit das Interesse an Technik geweckt habe, antworteten 46 Schüler/innen mit „ja“ bzw. „eher ja“. Fairerweise muss man dabei allerdings einen Höflichkeitsbias unterstellen.

Bei den beiden abschließenden Fragen konnten die Schüler/innen in offenen Antworten noch notieren, was sie besonders interessant fanden. Die nachfolgende Tabelle fasst die Ergebnisse aus den drei Grundschulklassen, unterschieden nach dem Thema der Einheit, zusammen:

Bilddarstellung/Grafik	Hardware	Codierung
<ul style="list-style-type: none"> - Die bunten Farben (6 mal) - Das Malen (4 mal) - Pixelgrafik – Vektorgrafik (4 mal) - Das Mischen der Farben (2 mal) - Das zum Fenster schauen [Anm. Spektroskop] - gar nichts 	<ul style="list-style-type: none"> - Alles war interessant (5 mal) - Das mit dem Computer auseinander nehmen (5 mal) - Für mich war das interessant, wo wir die Kabeln heraus nehmen durften (2 mal) - Die Festplatte (2 mal) - Die Hauptplatine (2 mal) - Der Kühler - Die Eingabe und die Ausgabe - Als ein paar Kinder Arbeitsspeicher gespielt haben - Die Speicherplatte war spannend - Als ich ein Teil vom Computer war - Zum Schluss, als ich das mit den Leitern gefragt habe - Die Striche, die der Professor erklärt hat [Anm. gemeint sind die Leiter am Motherboard] - Nichts 	<ul style="list-style-type: none"> - Das erste Spiel (6 mal) [Morsespiel] - Als wir mit Taschenlampen Signale gegeben haben [ebenfalls Morsespiel] - Unsere eigene Codierung (4 mal) - Der Morse Code (3 mal) - Die Codes und der Rechner des Computers - Alles - Das mit dem Baum [Codebaum] - Der Rechner

Bei Schüler/innen der Sekundarstufe I im Alter zwischen 13 und 15 Jahren ergab sich ein etwas anderes Bild. Sechs der insgesamt 17 Schüler/innen fanden das Thema Codierung interessant, vier wählten „eher ja“ und fünf „eher nein“. Zwei der Schüler/innen fanden das Thema überhaupt nicht interessant.

Das Interesse an Technik war geringer als bei den Grundschulern. Acht der 17 Schüler/innen interessierten sich für Technik, 9 nicht. Auch auf die Frage, ob die Einheit Interesse an der Technik geweckt hat antworteten nur 8 Schüler/innen mit ja. Die Rückmeldungen bei den freien Antworten ergaben:

- | | |
|-------------------------------|-------------------------|
| - Der Morsecode (6 Nennungen) | - Alle beide Stunden |
| - Der Codebaum (5 Nennungen) | - Die Huffman Codierung |
| - Dezimalsystem (2 Nennungen) | - Code erstellen |
| - Mir hat alles gefallen | - Das Spiel |

6 Resümee und Ausblick

Die Pilotphase an den Schulen hat gezeigt, dass die vorgestellten Einheiten im Großen und Ganzen für die Zielgruppe angemessen sind und Interesse bei den Schüler/innen wecken können. Die Rückmeldungen der Lehrer/innen sowie der Schüler/innen wurden von uns nebst eigenen Reflexionen in die Einheiten und Module eingearbeitet. Die überarbeiteten Einheiten werden bzw. sind auf der Website des Projektes publiziert, und stehen somit einem breiten Interessentenkreis aller Schulstufen zur Verfügung.

Im Jänner 2009 startete im Rahmen von *Forschung macht Schule* (ein Projekt des Bundesministeriums für Verkehr, Innovation und Technologie (bmvit)) eine dreistufige Kooperation zwischen der Universität Klagenfurt, heimischen Firmen sowie Kärntner Schulen aller Schulstufen. Die Kooperation dauert drei Semester, wobei jede teilnehmende Klasse pro Semester mindestens bei einer *Informatik-erLeben*-Einheit mitmachen wird. Weiters wird jede Klasse eine Firma besuchen, sowie an der Universität einen Blick in die Forschungslandschaft der Informatik wagen.

Literaturverzeichnis

- [AKLU07] Antonitsch, P., Krainer, L., Lerchster, R., Ukowitz, M.: Kriterien der Studienwahl von Schülerinnen und Schülern unter spezieller Berücksichtigung von IT-Studiengängen an Fachhochschule und Universität; IFF-Forschungsbericht, Universität Klagenfurt, März 2007.
- [ALS07] Antonitsch, P.K., Lassernig, U., Söllei, A.: Lehrarrangements in der Informatiklehrerbildung. In: Schubert, S. (Hrsg.): Didaktik der Informatik in Theorie und Praxis; Proc. INFOS 2007, LNI 112, Gesellschaft für Informatik, Bonn, 2007, pp. 91 – 99.
- [BWF06] Bell, T.; Witten, I. H.; Fellows, M.: Computer Science Unplugged. An enrichment and extension programme for primary-aged children.
<http://www.google.com/educators/activities/unpluggedTeachersDec2006.pdf> (12. 2. 2009).
- [DDG07] Dagys, V., Dagiene, V., Grigas, G.: Teaching Algorithms and Programming by Distance: Quarter Century's Activity in Lithuania. In: Dagiene, V., Mittermeir, R. (eds.): Information Technologies at School, Proc. 2nd ISSEP, TEV, Vilnius, 2006, pp. 402 – 412.
- [Kol08] Kolczyk, E.: Algorithm – Fundamental Concept in Preparing Informatics Teachers. In: Mittermeir, R.T., Syslo, M.M. (eds.): Informatics Education – Supporting Computational Thinking, Proc. 3rd. ISSEP, LNCS 5090, Springer, 2008, pp. 265 – 271.
- [Mich08] Micheuz, P.: Some Findings on Informatics Education in Austrian Academic Secondary Schools. In: Informatics in Education, 2008, Vol. 7(1), pp. 1 – 16; Institute of Mathematics and Informatics, Vilnius.
- [Mich09] Micheuz, P.: Theorie, Praxis und Pragmatik informatischer Bildung an der AHS; to appear in Proc. INFOS 2009.
- [RNH04] Reichert, R., Nievergelt, J., Hartmann, W.: Programmieren mit Kara: Ein spielerischer Zugang zur Informatik; Springer, 2004.
- [Reit05] Reiter, A.: Incorporation of Informatics in Austrian Education: The Project “Computer-Education-Society” in the School Year 1984/85. In: Mittermeir, R.T. (ed.): From Computer Literacy to Informatics Fundamentals, Proc. ISSEP 2005, LNCS 3422, Springer, 2005, pp. 4 – 19.

Untersuchungen zu informatischen Kompetenzen in Sachsen - eine empirische Studie

Katrin Büttner

Thomas Knapp

Steffen Friedrich

Mittelschule
„J. W. v. Goethe“

01809 Heidenau
katrin.buettner@arcor.de

Mittelschule
Kötzschenbroda

01445 Radebeul
msk-knapp@gmx.de

TU Dresden, Fakultät Informatik
AG Didaktik der Informatik

01062 Dresden
steffen.friedrich@tu-dresden.de

Shahram Azizi Ghanbari, Gregor Damnik, Jacqueline Pape, Franz Schott

TU Dresden, Fakultät Mathematik und Naturwissenschaften
Institut für Pädagogische Psychologie und Entwicklungspsychologie
01062 Dresden

schott@psychomail.tu-dresden.de

Abstract: Das Unterrichtsfach Informatik gehört seit 1992 zum Pflichtbereich in der Sekundarstufe I aller Schularten in Sachsen. Bei der Umsetzung der im Jahre 2004 neu erstellten Lehrpläne waren insbesondere in den Klassenstufen 7 und 8 neue Anforderungen zu realisieren. In einer Lernstandsanalyse, an der sich freiwillig 120 Schulen mit über 3000 Schülerinnen und Schülern beteiligt haben, sollte untersucht werden, mit welchem Ergebnis der Themenbereich "Klassen und Objekte" des Lehrplans an Mittelschulen und Gymnasien realisiert wurde und wie einzelne Kompetenzen ausgeprägt sind.

Orientiert an den Bildungsstandards Informatik und auf der Basis psychologischer Arbeiten zur Kompetenzdiagnostik wird damit ein Beispiel für empirische Untersuchungen zu informatischen Kompetenzen vorgelegt, das valide Daten liefert und hinsichtlich der Untersuchungsmethodik verallgemeinerbar ist.

1 Bildungsstandards und Informatikunterricht in Sachsen

Spätestens seit der Veröffentlichung der ersten Ergebnisse vergleichender empirischer Studien wie TIMSS oder PISA haben verschiedene Debatten und fast unzählige wissenschaftliche und journalistische Schriften das Interesse für Arbeiten unter dem Stichwort "Bildungsstandards" gestärkt. Möglicherweise ist auch dadurch ein reichlich diffuses Bild zu den Absichten und den wirklichen Veränderungen entstanden, die eine Orientierung an den Ergebnissen von Bildungsprozessen hervorbringen kann. Standardisierungen von Bildungszielen in der Schule existieren bereits in Form von Lehrplänen, zentralen Prüfungen, äußeren Strukturen von Schulen oder auch in Richtlinien der Verwaltung. Festlegungen dazu, was Schülerinnen und Schüler zu einem bestimmten

Zeitpunkt an Kenntnissen und Fertigkeiten erworben haben sollen, wurden bisher aus diesen Bildungszielen bestimmt. Wenn man wissen will, ob fachbezogene Kompetenzen tatsächlich vorliegen, benötigt man unterrichtspraktische Untersuchungen, die solche Festlegungen überprüfen können und vor allem einen Unterricht, in dem die Schüler die Chance haben, genau diese Kompetenzen zu erwerben.

In diesem Zusammenhang kann es nicht das Ziel von Bildungsstandards sein, eine zentralisierte innere Systematik von ausgewählten Schulfächern zu beschreiben. Solche Beschreibungen würden sich von bisherigen Zieltaxonomien kaum unterscheiden und für die Qualitätsentwicklung in der Bildung wenig bewirken. Die Darstellungen in Form von Bildungsstandards, auch die zur informatischen Bildung, sollten genau das beschreiben, worauf sich aufbauende Bildungsgänge oder die berufliche Praxis verlassen können. So, wie die Lesekompetenz nicht ausschließlich im Unterrichtsfach Deutsch entwickelt wird, ist auch für andere Kompetenzbereiche im jeweiligen fachlichen Kontext zu diskutieren, wo deren Herausbildung stattfindet. Ein Fachunterricht Informatik wird zur informatischen Bildung immer wesentliche Beiträge leisten, aber allein nicht ausreichen, um die jeweiligen Kompetenzen zu entwickeln. Bei der Erstellung von Bildungsstandards Informatik [BS07] wurden solche Inhalte und Methoden herausgearbeitet, die sich deutlich von denen anderer Fächer unterscheiden und in einem systematischen Unterricht im Fach Informatik geprägt werden sollten. Hinsichtlich der informatischen Kompetenzen kommt es folglich darauf an, dass deren Entwicklung auch von anderen Fächern unterstützt wird. Die Bildungsstandards Informatik beschreiben letztlich solche Kompetenzen, die das Minimum informatischer Bildung für alle Schüler erfassen und stellen somit Mindeststandards dar.

Wenn häufig insbesondere die Chancen von Bildungsstandards dargestellt werden, sollte der Blick auch auf mögliche Probleme gerichtet werden. So besteht bei einer überzogenen Orientierung auf Ergebnisse des Lernens die Gefahr, dass Unterricht zur Testvorbereitung mutiert und – auf eine automatische Ausprägung von Kompetenzen hoffend – nur die Lösung von typischen Aufgaben trainiert wird [KR05]. Andererseits kann eine Verbesserung von Qualität im Unterricht nicht allein durch die Einführung verbindlicher Standards erwartet werden. Es muss ein systematischer Veränderungsprozess eingeleitet werden, in dessen Gestaltung Lehrende von Beginn an einbezogen sind und zu deren Realisierung sie zeitliche Ressourcen, fachliche sowie didaktische Unterstützung erhalten. Gut eingepasste Lernstandsanalysen stellen eine Möglichkeit dar, diesen Prozess der Qualitätsentwicklung mit wissenschaftlicher Unterstützung zu fördern.

Solche Überlegungen sind für die Schulpraxis nur dann hilfreich, wenn es gelingt, in empirischen Untersuchungen zu solchen Daten zu gelangen, die auch Folgerungen zu Zusammenhängen von Aufgabenlösungen und Kompetenzen zulassen. Neben dem Vorliegen eines passenden Themengebietes sind ein auf gleicher Grundlage basierender Unterricht, eine solide fachliche Ausbildung der Lehrenden, eine genügend große Stichprobe und die vorliegende Genehmigung der Schulbehörde zur Durchführung von Untersuchungen mit Schülern notwendig. Gerade unter der Beachtung der Notwendigkeit eines ungestörten Unterrichts ist der letzte Aspekt verständlicherweise ein Grund, weshalb didaktisch-orientierte empirische Studien häufig eine relativ kleine Population

besitzen. Es bedarf der Beachtung vieler Facetten, um eine solche Lernstandsanalyse in Angriff nehmen zu können.

Mit Blick auf die genannten Aspekte bot es sich an, in Sachsen eine solche Untersuchung zu informatischen Kompetenzen durchzuführen. Seit 1992 existiert hier ein Pflichtfach Informatik in der Sekundarstufe I, das durch neue Lehrpläne im Jahre 2004 ausgebaut und inhaltlich modernisiert wurde [IB05]. In den Klassenstufen 7 und 8 sind in den Lehrplänen sowohl der Mittelschulen als auch der Gymnasien entsprechende Lernbereiche zur Modellierung von "Klassen" und "Objekten" vorhanden, die an Beispielen geeigneter Anwendungen thematisiert werden [LP04]. Ferner wurden in den letzten 20 Jahren über 600 Lehrer durch ein Studium für das Unterrichten des Faches Informatik in Form eines berufsbegleitenden Studiums (einschl. Staatsprüfung) qualifiziert. Das aktive Mitwirken sächsischer Lehrer bei der Erarbeitung der Bildungsstandards Informatik, in Fort- und Weiterbildungen (auch für andere Fächer) und insbesondere der systematische Fachunterricht führten zu einer Aufgeschlossenheit für das Fach Informatik. So konnte nach Genehmigung durch das Sächsische Staatsministerium für Kultus eine Lernstandsanalyse für Schüler der 8. Klassen auf freiwilliger Basis geplant und durchgeführt werden.

2 Zur Durchführung einer Lernstandsanalyse in Sachsen

Im Schuljahr 2007/08 haben die ersten Schülerinnen und Schüler, die vollständig nach den neuen Lehrplänen unterrichtet wurden, die Schule verlassen. Spätestens zu diesem Zeitpunkt entstand die Frage, in welcher Qualität die Umsetzung dieser Lehrpläne geschieht. Die Idee, dies genauer zu untersuchen und dabei zu prüfen, welches vergleichbare Niveau die Schülerinnen und Schüler in einem relativ eng begrenzten Inhaltsbereich erzielen, entstand bei einem Treffen der Kolleginnen und Kollegen des „Stammtisch Informatik-Mittelschule“ in Dresden. Eigentlich sollten zunächst nur die Leistungen der Schülerinnen und Schüler der eigenen Schulen mittels gemeinsam erarbeiteter Aufgaben analysiert werden. Um eine solche Untersuchung zu initiieren war zu klären, welcher Themenbereich und welche Aufgaben von allen akzeptiert werden. Weiterhin war zu bestimmen, wie daraus Rückschlüsse zur Erreichung der Lehrplanziele abgeleitet werden könnten und welche Hilfe bei der Auswertung benötigt wird.

Nach Veröffentlichung der Bildungsstandards Informatik [BS07] war es folgerichtig, dass sich dieses Vorhaben dort anlehnen sollte. Es galt also, genau solche Aufgaben zu formulieren, die Mindestanforderungen an die informatische Bildung darstellen und damit in den Bildungsstandards formulierte Kompetenzen widerspiegeln.

Ausgehend von diesen Zielen wurden auf Initiative der Kollegen von Mittelschulen und mit späterer Mitarbeit aus Gymnasien Aufgaben einer Lernstandsanalyse Informatik entwickelt. Bei nahezu identischen Lehrplaninhalten im Themenbereich „Klassen und Objekte“, der zur Untersuchung ausgewählt wurde, sollten die Ergebnisse auch zwischen den Schultypen verglichen werden. Um den rechtlichen Rahmen abzusichern, wurde schließlich unter der Federführung der GI-Fachgruppe "Informatische Bildung in Sachsen und Thüringen" eine Genehmigung für die Durchführung der Lernstandsanalyse

beim Sächsischen Staatsministerium für Kultus beantragt, die unter der Bedingung der Freiwilligkeit der teilnehmenden Schulen erteilt wurde.

Die Erstellung der gesamten Aufgaben erfolgte in mehreren Zusammenkünften im Zeitraum eines halben Jahres. Ein Vortest wurde von den beteiligten Kolleginnen und Kollegen an deren Schulen durchgeführt, um ein erstes Feedback zu erhalten. Danach konnten die Aufgaben abschließend formuliert sowie Erwartungsbilder und Bewertungsmaßstäbe abgestimmt werden. Die Auflage, dass die Lernstandsanalyse freiwillig durchzuführen sei, erforderte einmal zusätzlichen Aufwand hinsichtlich der Motivation der Informatiklehrer. Zum Anderen war der Ablauf so zu organisieren, dass der Aufwand bei der Durchführung sowie der Erfassung der Resultate minimiert wird. Über die unterschiedlichsten Kontaktmöglichkeiten, wie die GI-Fachgruppe Sachsen/Thüringen, Absolvententreffen an der Technischen Universität Dresden oder Briefe der Fachberater Informatik an die Kollegen ihrer Regionalgruppen, sollten möglichst viele Lehrer aller Schularten zur Teilnahme an der Lernstandsanalyse gewonnen werden.

Für die Auswahl der Aufgaben standen die Forderungen der Lehrpläne als Voraussetzung. In den Lehrplänen zum Fach Informatik (Abb. 1) der Mittelschule bzw. der Gymnasien sind die folgenden Ziele formuliert:

Lehrplan Mittelschule	Lehrplan Gymnasium
Klasse 7	
Kennen grundlegender Datenstrukturen in einer ausgewählten Anwendung - Begriffe: Objekt, Attribut, Attributwert, Operation - Zuordnung von konkreten Objekten zum Modell Objekt – Attribut – Attributwert - eine einfache Darstellungsform	Beherrschen typischer Handlungen bei der Nutzung von Anwendungen - Objekte auswählen - Attribute festlegen - Methoden nutzen
Klasse 8	
Kennen von Klassen - Begriffe: Klasse, Attribute, Attributwertebereiche, Methoden - Klassen aus Erfahrungswelt; - Klassen aus der Informatik, Beherrschen der Zuordnung zwischen Objekten und Klassen	Anwenden typischer Handlungsfolgen zum Repräsentieren von Informationen - Abbildung von Informationen als Daten unter Verwendung von Modellen - Klassen und Objekte

Abb. 1: Auszüge aus den Lehrplänen Informatik [vgl. LP04]

Lernstandsanalyse – Informatik Klasse 7/8 Punkverteilung

Teil 3 – vom Objekt zur Klasse
 Der Sportverein „Fit4School“ möchte als Zeichen seiner Zusammengehörigkeit für jeden Teilnehmer ein T-Shirt mit Namen und Logo bedrucken zu lassen.
 Kataloge wurden gewälzt und die folgenden Angebote gefunden.

		
XL Baumwolle/Elastan 11,00€	M Baumwolle 7,50€	M Microfaser 8,00€

Bei genauem Lesen stellt der Trainer fest, dass die Shirts auch in XS, S, L und in grün, blau, lila und pink lieferbar sind.

1. Stelle die Klasse zu den abgebildeten Objekten in einer dir bekannten Form dar. /5

Abb. 2: Beispielaufgabe

Die dargestellte Aufgabe (Abb. 2) ist die letzte der Lernstandsanalyse und bezieht sich auf die oben genannten Lehrplaninhalte. Mit ihr soll das im Unterricht erreichte Abschlussniveau in der Klassenstufe 8 erfasst werden. Die Bewertungseinheiten sollten dabei auf die korrekte (vom Lehrer vermittelte) Darstellungsform, den Namen der Klasse, mindestens ein Attribut, den Attributwertebereich und das Hinzufügen einer Methode erteilt werden.

Das Erwartungsbild: Die Schüler erkennen, dass mehrere Objekte derselben Klasse dargestellt sind und finden durch Verallgemeinern einen Namen für die Klasse. Sie erkennen die Attributwerte und können das zugehörige Attribut benennen. Beim Darstellen der Klasse suchen sie nach weiteren Attributwerten und fassen diese in einem Attributwertebereich zusammen. Die Schüler wissen, dass beim Beschreiben einer Klasse auch Methoden benannt werden und finden diese selbstständig. Alle Begriffe werden von den Lernenden in einer von ihnen selbst gewählten Form korrekt dargestellt.

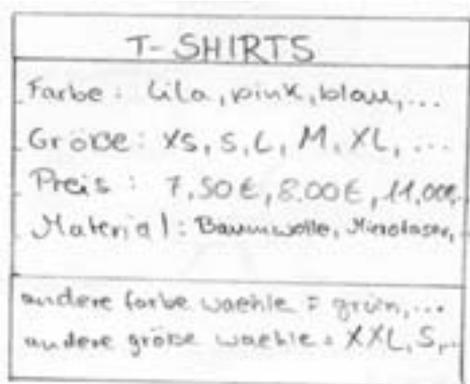


Abb. 3: Beispiel einer Schülerantwort

Bei ersten Auswertungen konnte bereits festgestellt werden: Das Finden des Klassenbegriffs bereitete den Schülern Schwierigkeiten. In einem Teil der Ergebnisse (z.B. Abb. 3) ist zu erkennen, dass dabei der letzte Schritt zur Verallgemeinerung fehlt. Die Bewertung der „korrekten Darstellungsform“ wurde durch die Lehrenden unterschiedlich interpretiert und ist wahrscheinlich nicht vergleichbar. Da die Methoden im Text nicht ausdrücklich gefordert wurden, sind diese nur von wenigen Schülern genannt worden. Das kann vielfältige Gründe haben, wie zum Beispiel, dass sich die Schüler wortgenau an die Aufgabenstellungen gehalten haben oder auch nicht gelernt haben, Methoden mit Klassen in Zusammenhang zu bringen. Es ist auch denkbar, dass die Aufmerksamkeit der Schüler zum Ende der Lernstandsanalyse nachgelassen hat.

Für die Durchführung der Lernstandsanalyse mit den Schülerinnen und Schülern wurde ein Zeitraum von zwei Wochen vereinbart. Im Anschluss erfolgte die Online-Erfassung der Schülerergebnisse durch die teilnehmenden Lehrer. Um für die spätere Auswertung auch Details zur Verfügung zu haben, war es wichtig, jede erreichte Bewertungseinheit einzeln zu erfassen.

Eine besondere Unterstützung bei der Durchführung bot die Pädagogische Plattform des Sächsischen Bildungsservers [SBS09]. Auf entsprechenden Seiten standen die Informationen über das Anliegen dieser Analyse sowie Aufgaben und Erwartungsbilder zum Download zur Verfügung. Mit Unterstützung einer wissenschaftlichen Abschlussarbeit [WI08] konnte auch die Online-Erfassung vorbereitet und effektiv durchgeführt werden. Sie erfolgte ebenfalls über die Pädagogische Plattform. Die Zusammenfassung der Ergebnisse der jeweiligen Schule wurde dem Eingebenden sofort nach dem Abschluss der Eingabe als Datei zur Verfügung gestellt. Außerdem ist durch eine Übersicht ein Vergleich zwischen den Gesamtergebnissen (Abb. 4) und den Ergebnissen der jeweiligen Schule möglich. Hier sind zu allen Teilaufgaben die durchschnittlich erreichten Punkte und Anteile an den maximal zu erreichenden Punkten angegeben.

Alle Schulen: (3406 Schüler)

	1.1	1.2	1.3	∑ Teil 1	2.1	2.2	∑ Teil 2	Teil 3	gesamt
Ø	3.3	3.6	2.3	9.2	1.8	1.7	3.4	2.5	15.1
%	83	90	75	84	59	56	57	49	69

Abb. 4: Auszüge aus einer ersten Auswertung

Dank dieser vielfältigen Bemühungen haben sich insgesamt 118 Schulen an dieser Lernstandsanalyse beteiligt. Von diesen Schulen waren 90 Mittelschulen, das sind 26 % aller Mittelschulen in Sachsen. Von 149 Gymnasien haben 27 teilgenommen, das sind 18 %. Es ist gelungen, die Resultate von insgesamt 3406 Schülerinnen und Schülern zu erfassen, die im Schuljahr 2008/09 im Pflichtfach Informatik in den 8. Klassen unterrichtet wurden. Im Vergleich zu vielen anderen Untersuchungen zur Diagnostik von Kompetenzen zeichnet sich diese Lernstandsanalyse somit durch einen großen Stichprobenumfang und eine vergleichsweise hohe Repräsentativität der Daten aus.

Aufgrund der großen Datenmenge kann eine tiefgründigere Analyse erst schrittweise erfolgen. Insbesondere durch Nutzung von untersuchungsmethodischen Erfahrungen aus dem Bereich der Kompetenzdiagnostik ergeben sich weitere Zusammenhänge, die in bisherigen Untersuchungen zu informatischen Kompetenzen noch nicht berücksichtigt wurden. Als Ergebnis sind später auch Hinweise für kompetenzorientierte Aufgabenformulierungen und Tipps zur Umsetzung der Lehrplaninhalte in einzelnen Unterrichtssituationen zu erwarten.

3 Kompetenzdiagnostik am Beispiel informatischer Bildung

3.1 Kompetenzorientierter Unterricht

Wenn es um eine zweckmäßige Bestimmung von Kompetenzen in der Bildung geht, ist zunächst klarzustellen, welcher Zweck verfolgt wird. Nach dem „PISA-Schock“ soll durch eine verstärkte „Output-Orientierung“ eine verbesserte Qualität von Schule und Unterricht erreicht werden. Mit „Output“ werden dabei die bei den Lernenden vorzufindenden Unterrichtsergebnisse bezeichnet.

Dies wirft folgende Fragen auf, deren Beantwortung nicht isoliert betrachtet werden darf:

- Wie sind Unterrichtsergebnisse bei den Lernenden zu erfassen?
- Wie kommen Unterrichtsergebnisse bei den Lernenden zustande?
- Was führt zu besseren Unterrichtsergebnissen?

Die Beantwortung der ersten Frage betrifft die Analyse von Leistungen der Lernenden, die heute „Kompetenzdiagnostik“ genannt wird. Die Behandlung der zweiten Frage zeigt, dass neben einem zielorientierten Unterricht noch andere Faktoren die Unterrichtsergebnisse beeinflussen, z.B. Leistungsdispositionen. Bei Überlegungen zur dritten Frage wird klar, dass eine Rückmeldung der bisher erreichten Ergebnisse – wie bei jeder Qualitätssicherung – ein grundlegender Faktor ist [SG08]. Die vorliegende Lernstandsanalyse zur informatischen Bildung betrifft insbesondere die erste Frage. Im Folgenden soll daher eine Strategie zur Auswertung der Daten der Lernstandsanalyse skizziert werden, die auch auf andere Untersuchungen übertragen werden kann. Aus den Ergebnissen der durchgeführten Analysen sollen Aussagen über die Ausprägung der informatischen Kompetenzen bei den Schülerinnen und Schülern sowie Hinweise zur Optimierung der in der Lernstandsanalyse verwendeten Aufgaben und der Lehre im Fach Informatik im Allgemeinen abgeleitet werden.

3.2 Ein semantisches Netz zur Analyse der Aufgaben der Lernstandsanalyse

Um die Struktur der informatischen Kompetenzen, die von den Schülerinnen und Schülern im Unterricht erworben werden sollen, besser zu verstehen und die didaktische Gestaltung des Informatikunterrichts noch stärker als bisher auf die Ausprägung und Verfeinerung dieser Kompetenzen auszurichten, bieten sich psychologische Aufgaben-

analysen an. Im Zuge solcher Analysen können die mentalen Modelle ermittelt werden, die bei den Schülern entstehen sollten, nachdem sie kompetenzorientiert nach dem vom Lehrplan vorgegebenen Stoff unterrichtet wurden.

Das mentale Modell eines Schülers zum objektorientierten Ansatz könnte wie folgt durch ein semantisches Netzwerk dargestellt werden (Abb. 5).

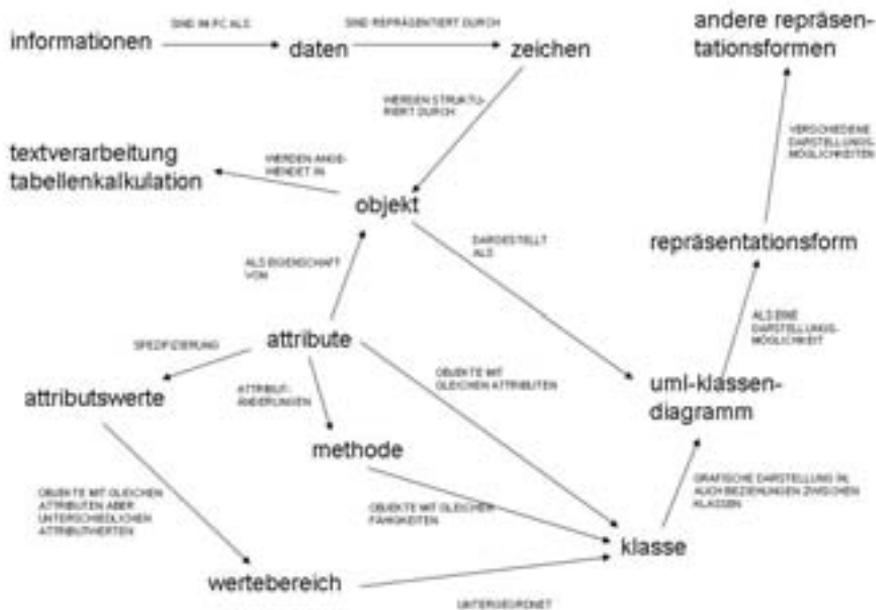


Abb. 5: Beispiel für semantisches Netz

In diesem semantischen Netzwerk sind verschiedene Elemente und Relationen erfasst, die für diesen Schüler dann auch in Anwendungssituationen verfügbar sind. Die Elemente des Netzwerks – wie zum Beispiel „Methoden“, „Klassen“ oder auch die „Objektorientierung“ – sind nicht nur bloße Wörter, sondern beziehen sich auf Konzepte, die der Schüler mit der Unterstützung seines Lehrers erworben hat. Er kann somit diese Begriffe nicht nur einordnen, sondern auch mit selbstgewählten Beispielen untersetzen, sie auf unterschiedliche Aufgaben anwenden und miteinander in Beziehung setzen. Einen Vorschlag, wie solche Beziehungen zwischen Begriffen aussehen könnten, machen die hier mit Pfeilen und in Großbuchstaben dargestellten Relationen deutlich. Im Beispiel bedeutet das, dass der Schüler in der Lage ist zu erklären, dass Objekte aus der Klasse „Zeichen“ (wie beispielsweise ein ganz konkreter Buchstabe) zugeordnete Attribute (wie zum Beispiel eine „Farbe“ oder auch eine „Größe“) haben und diese wiederum einen spezifischen Attributwert zugeschrieben bekommen (etwa „rot“ oder Schriftgröße „10“). In dem in Abbildung 5 dargestellten semantischen Netzwerk wird

ferner berücksichtigt, dass der Schüler außerdem verschiedene Darstellungsformen und Anwendungsbereiche kennen und nutzen sollte.

Für die Erstellung eines solchen Modells bleibt im Moment der Lehrplan die wesentliche Orientierungsgrundlage. Darin implizit dargestellte und am Ende des Lernbereichs zu erwartende Kompetenzen müssen erst sukzessive entwickelt und sicher auch getestet werden. Gleichzeitig entstehen im Rahmen dieser psychologischen Analysen auch Vorschläge, wie der Lehrstoff künftig in das Vorwissen integriert werden könnte. So müsste beispielsweise darüber nachgedacht werden, ob über einen altersgemäß formulierten Informationsbegriff eine Einführung nicht günstiger gestaltet wird. Es sollte auch geprüft werden, ob Schüler durch das Kennenlernen weiterer Repräsentationsformen (wie z.B. die Punktnotation) besser darauf eingestellt sind, das mentale Modell zu erweitern und zu verfeinern.

Die mentalen Modelle, die aus den Angaben des Lehrplans abgeleitet und erstellt werden, können auch dabei helfen, die (Inhalts-)Validität der Aufgaben der Lernstandsanalyse zu beurteilen und zu optimieren. Hierfür sollten zunächst auch für die Aufgaben der Lernstandsanalyse semantische Netzwerke erstellt werden. Im Anschluss kann dann geprüft werden, inwiefern die Struktur dieser Netzwerke den mentalen Modellen entspricht, die für die im Lehrplan umrissenen informatischen Kompetenzen ermittelt wurden.

3.3 Empirische Untersuchungen zu informatischen Kompetenzen

Um die Daten der Lernstandsanalyse zum Fach Informatik interpretieren zu können, sind die Auswertungen entsprechend aktueller Testanalysemethoden vorzunehmen. Dazu bieten sich verschiedene Verfahren an, die in der nächsten Zeit zur Auswertung herangezogen werden. Die Grundlage hierfür bilden die von den Schülerinnen und Schülern in der Lernstandsanalyse erreichten Testrohwerte.

Zunächst sollen mittels Korrelationsanalysen die statistischen Zusammenhänge der einzelnen Testaufgaben ermittelt werden. Darauf aufbauend wird mit Hilfe von Faktorenanalysen die Dimensionalität des Tests untersucht. Im Anschluss daran sollen die Aufgaben einer empirischen Itemanalyse nach der Klassischen Testtheorie (KTT) unterzogen werden. Im Zuge dessen werden die Schwierigkeitsindizes und Trennschärfen der Aufgaben bestimmt. Entsprechend der Konventionen von Lienert und Raatz [LR94], denen zufolge Items außerhalb des Schwierigkeitskorridors zwischen 0,2 und 0,8 nicht Bestandteile einer Testendform sein sollten, werden „zu leichte“ und „zu schwere“ Aufgaben aus den resultierenden KTT-Skalen entfernt. Als Maß der internen Konsistenz der ermittelten Skalen wird der Alpha-Koeffizient nach Cronbach berechnet [LR94], [RO96]. Ferner sollen unter Verwendung des Programmpakets ‚Irm‘ für die Statistiksoftware R [RI06] weitere Itemanalysen nach dem Rasch- und nach dem Birnbaummodell durchgeführt werden. Im Vordergrund steht hierbei vor allem die Analyse der Höhe der Abweichungen zwischen den tatsächlichen Lösungshäufigkeiten der Schüler und den Lösungshäufigkeiten, die anhand der Modelle ermittelt werden. So können größere Abweichungen darauf hinweisen, dass einzelne Aufgaben der Lernstandsanalyse einer Optimierung bedürfen.

Im Ergebnis der Dimensionalitätsprüfungen und der empirischen Itemanalysen entstehen mehrere Testskalen, auf denen die Ergebnisse der Schüler abgebildet und miteinander verglichen werden können. Die ermittelten Skalenwerte bilden außerdem die Grundlage für weitere statistische Auswertungen. So werden die Unterschiede in den Skalenwerten zwischen einzelnen Teilstichproben (z.B. Gymnasium und Mittelschule) mit Hilfe von Regressions- und Varianzanalysen statistisch untersucht.

4 Ausblick

Mit der dargestellten Lernstandsanalyse ist es gelungen, Daten zur informatischen Bildung einer sehr großen Anzahl von Schülern, die einen Pflichtunterricht im Fach Informatik in der 8. Jahrgangsstufe besuchen, zu erfassen und unter kompetenzdiagnostischem Aspekt auszuwerten. Ausgehend vom Wunsch der Lehrenden, ihre bei der Umsetzung eines neuen Lehrplans erreichten Ergebnisse zu vergleichen, wurde damit gleichzeitig untersuchungsmethodisch Neuland beschritten. Neben einer relativ großen Population von Schülern, die an der Lernstandsanalyse teilgenommen haben, wurde die gesamte Untersuchung von der Bereitschaft zur Teilnahme bis zur Datenerfassung und -auswertung über die Pädagogische Plattform des Sächsischen Bildungsservers abgewickelt. Beispielsweise konnten Lehrer nach Abschluss der gesamten Dateneingabe für alle Teilnehmer ihren Datensatz mit dem Gesamtdurchschnitt vergleichen. Damit ist ein Verfahren getestet und verallgemeinerbar, das auch in anderen Fächern sofort genutzt werden kann.

Erst eine detailliertere Auswertung kann zeigen, welche Zusammenhänge zu den Mindeststandards bestehen und inwieweit diese Folgerungen zur Gestaltung eines kompetenzorientierten Unterrichts bzw. angepasster Testszenarien gestatten. Bereits jetzt wird deutlich, dass es der Entwicklung und Bereitstellung von Aufgaben bedarf, die an Kompetenzen der Bildungsstandards orientiert sind und Lehrende bei der Planung und Gestaltung von Unterricht unterstützen. Die Erarbeitung eines Trainingsprogramms zur Ausarbeitung geeigneter Aufgaben ist folgerichtig ein geplanter nächster Schritt.

Literaturverzeichnis

- [BS07] Grundsätze und Standards für die Informatik in der Schule - Bildungsstandards Informatik für die Sekundarstufe I. Gesellschaft für Informatik e.V., Bonn In: LOG IN 150/151 (2008)
- [IB05] Autorenteam: Informatische Bildung im Freistaat Sachsen. In: Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung – INFOS'05. Lecture Notes in Informatics (LNI), P-60, S. 11-26
- [KR05] Krumm, H.-J.: Hilfreiche Standardisierung oder fatale Normierung. Gedanken zur Problematik von Bildungsstandards und Lernstandserhebungen. In: Bausch, K.-R.; Burwitz-Melzer, E.; Königs F. G. & Krumm H.-J. (Hrsg.). Bildungsstandards für den Fremdsprachenunterricht auf dem Prüfstand, S. 151-158

- [LP04] Lehrplan Informatik für Mittelschulen und für Gymnasien unter:
<http://www.sachsen-macht-schule.de/apps/lehrplandb/> (01.03.2009)
- [LR94] Lienert, G.A. & Raatz, U.: Testaufbau und Testanalyse (5. Aufl.). Weinheim: Beltz, PVU,1995.
- [RO96] Rost, J.: Testtheorie Testkonstruktion. Bern: Huber, 1996
- [RI06] Rizopoulos, D. ltm: An R package for latent variable modelling and item response theory analysis. Journal of Statistical Software, 17 (2006) 5, S. 1-25.
- [SBS09] Pädagogische Plattform des Sächsischen Bildungsservers
<http://www.sn.schule.de/lisa/> (01.03.2009)
- [SG08] Schott, F & Azizi Ghanbari, S.: Kompetenzdiagnostik, Kompetenzmodelle, kompetenzorientierter Unterricht. Zur Theorie und Praxis überprüfbarer Bildungsstandards ComTrans - ein theoriegeleiteter Ansatz zum Kompetenztransfer als Diskussionsvorlage. Waxxmann Verlag, 2008
- [WI08] Windisch, M.: Untersuchung zur informatischen Bildung an Schulen Sachsens – die Bildungsstandards und der Lehrplan: Vergleich, Interpretation, Kopplung und Umsetzbarkeit. Wissenschaftliche Arbeit zum Staatsexamen. TU Dresden, 2008

Beitrag der informatischen Bildung zu Schlüsselkompetenzen am Beispiel Internetworking

Kirstin Schwidrowski

Didaktik der Informatik und E-Learning

Universität Siegen

Hölderlinstr. 3

57076 Siegen

schwidrowski@die.informatik.uni-siegen.de

Abstract: Die Wissensgesellschaft stellt jedem in den verschiedenen Lebensbereichen komplexe Anforderungen. Die OECD hat im Rahmen des DeSeCo-Projektes einen konzeptuellen Referenzrahmen für die Bestimmung und Messung von Schlüsselkompetenzen entwickelt, mit denen Individuen in der heutigen Gesellschaft bestehen und diese mitgestalten können. Der Themenbereich Internetworking bietet für Lehr-Lernprozesse Anknüpfungspunkte für den Erwerb von Schlüsselkompetenzen. Auf nationaler Ebene sind mit den Bildungsstandards für die Sekundarstufe 1 Kompetenzen bestimmt worden, die einen Beitrag zu Schlüsselkompetenzen leisten.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21.–24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Informatische Bildung im Primarbereich

Henry Herper, Volkmar Hinz

Institut für Simulation und Graphik
Otto-von-Guericke-Universität Magdeburg
Postfach 4120
39016 Magdeburg
henry.herper@ovgu.de
volkmar.hinz@ovgu.de

Abstract: Die aktuellen Entwicklungen zeigen, dass Kinder heute immer früher mit Computern in Berührung kommen. Darauf muss auch im Rahmen der schulischen Bildung reagiert werden. Kinder können schon in der frühkindlichen Erziehung an die Nutzung des Computers herangeführt werden. Am Beispiel verschiedener Initiativen, wie „One Laptop per Child“ und „Klassenzimmer der Zukunft“ wird gezeigt, welche Möglichkeiten sich für die Bildung in der Grundschule ergeben, wenn jedes Kind mit einem persönlichen Laptop ausgestattet ist. Einige Möglichkeiten zum Erwerb der informatischen Kompetenzen in Verbindung mit Fachkompetenzen werden vorgestellt. Abschließend werden die veränderten Anforderungen an die IT-Schulinfrastruktur aufgezeigt und ein möglicher Lösungsansatz vorgestellt.

1 Ausgangssituation

Informatische Bildung ist heute unbestritten eine Kulturtechnik, deren Kompetenzen nicht nebenbei erworben werden können. Wann sollte mit der informatischen Bildung begonnen werden? In „Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen“ der GI e.V. von 2000 beginnt die informatische Bildung in der Primarstufe. Darin heißt es: „Die erste Begegnung mit Informatiksystemen in der Primarstufe muss pädagogisch und fachlich sehr behutsam und verantwortungsbewusst gestaltet werden. Zunächst intuitiv – aber fachlich korrekt – sollten im vorfachlichen Unterricht beim Einsatz interaktiver Informatiksysteme als Werkzeug und Medium sowohl erste Grundfertigkeiten im Umgang mit Informatiksystemen erworben als auch, dem Alter der Schülerinnen und Schüler angemessen, erste Grundkenntnisse dazu vermittelt werden.“ [GI00]

In den letzten 10 Jahren hat sich auch auf diesem Gebiet ein Wandel vollzogen. Die Kinder haben früher Kontakt zum Medium Computer und der Computer wird als Spielgerät und Werkzeug angesehen. Auch in der frühkindlichen Erziehung in einigen Kindergärten haben Computer Einzug gehalten. Im Rahmen der Microsoft-Bildungsinitiative „Schlaumäuse – Kinder entdecken Sprache“ wurde 2003 Lernsoftware für die frühkindliche Sprachförderung entwickelt [KO06]. Zielgruppe sind Kinder im Alter von drei bis sieben Jahren im Vorschulbereich. Diesem Projekt liegt das Konzept des Entfaltenden Lernens zugrunde.

Der Computer wurde in diesem Projekt als Medium eingesetzt. Der Erwerb von informatischen Fertigkeiten orientiert sich an dem, was zur Nutzung der Software notwendig ist. Die Auswertung einer begleitenden Studie zur Einführung des Schlaumäuse-Projektes zeigte, dass die Kinder in sehr kurzer Zeit die notwendigen Fertigkeiten zum Umgang mit dem Computer und zur Nutzung der Software entwickelten. Bemerkenswert ist, dass etwa 50% der Kinder dieser Altersstufe schon durch die Nutzung von Computern im häuslichen Bereich über Grundkenntnisse verfügen. Die Studie zeigt auch, dass ein erheblicher Teil der Kompetenzen durch das Lernen der Kinder voneinander erworben wird. Im Rahmen dieser frühkindlichen Medienbildung lernen die Kinder spielerisch den Computer als nützliches Werkzeug kennen und seine Grundfunktionen zielgerichtet zu nutzen.

Die Studien im Rahmen des „Schlaumäuse-Projektes“ [KO06] zeigten, dass unterschiedliche Ansätze zur Nutzung des Computers durch die Kinder im Kindergarten realisierbar sind. Eine Möglichkeit besteht darin, einen Computerarbeitsplatz im Gruppenraum zu installieren und den Kindern einen selbstbestimmten, ständigen Zugriff auf diesen Arbeitsplatz und seine Software anzubieten. Bei dieser Form ist es schwierig, die Nutzungszeit der Computer durch die Kinder zu regulieren. Die maximale Nutzungszeit sollte 30 Minuten nicht wesentlich überschreiten und Kinder sollten die Möglichkeit haben, die Nutzung jederzeit zu beenden. Ein anderer Ansatz besteht darin, einen Computer oder mehrere Computerarbeitsplätze im Rahmen eines Angebotes zur Verfügung zu stellen und gezielt Themen zu bearbeiten. Angebot bedeutet in diesem Fall, dass die Kinder entsprechend ihrer Interessen aus verschiedenen Beschäftigungsangeboten eines auswählen können. Die Dauer eines solchen Angebotes liegt zwischen 60 und 90 Minuten. Es wird ein bestimmter Themenbereich oder eine Aufgabenstellung spielerisch angeleitet bearbeitet. Auch in diesem Fall sollte die direkte Arbeitszeit am Computer auf ca. 30 Minuten begrenzt werden.

Im Rahmen dieser Computernutzung erlernen die Kinder Grundfertigkeiten bei der Nutzung des Computers. Die Eingabeaktionen werden durch Tastatur, Maus, Touchscreen oder Tablett realisiert. Für die Ausgabe werden der Bildschirm und der Audio-Ausgang genutzt. Auch der Aufbau einfacher Steuerungen mit einem geeigneten Baukastensystem, wie z.B. Lego-Education-WeDo ist mit Kindern dieser Altersklasse möglich.

Durch den Einsatz von Computern in der frühkindlichen Erziehung wird das Bildungsangebot erweitert. Das Ziel besteht nicht darin, z.B. den klassischen Buntstift und Malpinsel durch ein Malprogramm zu ersetzen, sondern den Kindern ein weiteres Gestaltungsmittel anzubieten. Kinder die in diesem Bereich erste Erfahrungen mit der Nutzung des Computers gemacht haben, werden diesen zukünftig zielgerichtet als Werkzeug zur Unterstützung der Problemlösung nutzen.

Einen wesentlichen Beitrag zur Einführung von Computern im Bildungsbereich der Grundschule leistete das von Nicholas Negroponte 2005 vorgestellte Projekt „One Laptop Per Child (OLPC)“. Im Vorfeld durchgeführte Untersuchungen haben gezeigt, dass klassische PCs weder von der Hardware noch von der Software für Kinder im Grundschulalter optimal geeignet sind. Daher wurde im Rahmen dieser Initiative ein speziell für die Bildung geeigneter Laptop entwickelt, der kindgerecht ist und zu einem günstigen Preis beschafft werden kann. Dieser wird entsprechend seiner Kennzeichnung auch als XO oder 100\$-Laptop bezeichnet. Mit dem OLPC-Projekt soll der digitalen Kluft zwischen Industrie- und Schwellenländern entgegengewirkt werden.

Die Besonderheit dieses Projektes besteht darin, nicht die Schulen mit Computertechnik auszustatten, sondern jedem Schüler ein persönliches Notebook zu geben. Diese Form der Arbeit mit dem Computer wird auch als 1:1-Lernen bezeichnet. Das Vertriebskonzept ist nicht auf den individuellen Erwerb ausgerichtet, sondern darauf, dass ganze Schulen bzw. Regionen über Landesverträge mit der entsprechenden Technik ausgestattet werden. Auch unterscheidet sich der Einführungszeitpunkt von der bisher typischen Vorgehensweise. Die Schülerinnen und Schüler erhalten den XO in der Grundschule.

Das technische Konzept des XO beinhaltet viele Neuerungen. Er ist mit einer kindgerechten Tastatur ausgerüstet und verfügt über ein eigens entwickeltes Display, das auch bei direkter Sonneneinstrahlung noch lesbar ist. Das integrierte WLAN-Interface mit dem Mesh-Netzwerkkonzept ermöglicht die einfache Vernetzung der Computer untereinander einschließlich einer intelligenten Erweiterung der Reichweite und die Verbindung zum Internet. Bei der Auswahl der Software wurden ausschließlich Open-Source-Entwicklungen verwendet. Der Computer wird in seiner Grundkonfiguration schon mit vielen Programmen für unterschiedliche Anwendungsgebiete geliefert.

Bei der Benutzungsoberfläche wurde mit „Sugar“, welches auf das Betriebssystem LINUX aufsetzt, ein neuer Weg gegangen. Die graphische Benutzungsoberfläche ist eine Lernumgebung und bildet nicht, wie viele traditionelle Benutzungsoberflächen, einen Schreibtisch nach. Der Anwender wählt bei „Sugar“ eine Aktivität aus und kann damit arbeiten. Als Aktivität wird ein Programm mit den zugehörigen Daten bezeichnet. Die auf klassischen PC vorherrschende ordnerbasierte Verzeichnisstruktur wurde durch ein aktivitätsbezogenes Journal ersetzt. In diesem sind die Einträge nach zugeordneten Aktivitäten und zeitlich geordnet. Die Aktivitäten können von mehreren Nutzern gleichzeitig bearbeitet werden. Das kooperative Arbeiten im Mesh-Netzwerk wird durch die Sugar-Oberfläche direkt unterstützt.

Damit ist der XO ein Computer, der als Medium im Unterricht Lehrbücher ersetzen und ergänzen kann, das kooperative Arbeiten untereinander unterstützt, als Kommunikationsmittel dient und den Zugang zum Internet ermöglichen kann. Die vorinstallierten Aktivitäten unterstützen das Lernen in verschiedenen Unterrichtsfächern. Mit der integrierten Kamera können Bilder für multimediale Applikationen erfasst werden. Der XO kann als Computer in der informatischen Grundbildung eingesetzt werden. Aktivitäten, die eine erste Erstellung von Programmen in unterschiedlicher Form ermöglichen sind z.B. Etoys, Scratch und Python. Die Aktivität Etoys stellt einen umfangreichen Medienbaukasten zur Verfügung, mit dem die Kinder ihre Ideen umsetzen können, indem sie Geschichten visualisieren und animieren.



Abbildung 1: XO - Computer

Mit dem OLPC-Projekt hat Nicholas Negroponte 2005 nicht nur die Entwicklung des XO begründet, sondern viele andere Hersteller dazu bewegt, an der Entwicklung von kostengünstigen Notebooks, die für Kinder im Schulalltag geeignet sind, zu arbeiten. Auch wurde ein grundlegender Konzeptwandel für die informatische Bildung an Schulen eingeleitet. Das bis heute noch in fast allen Schulen dominierende Computerlabor mit eingeschränkter Verfügbarkeit und meist hohem organisatorischen Aufwand zur Nutzung im Fachunterricht wird zukünftig an Bedeutung verlieren. Das persönliche Notebook, eingebunden in eine entsprechende Schul-Infrastruktur, wird ein normales Arbeitsmittel in jedem Unterrichtsfach.

Dieses Konzept, heute noch als „Klassenzimmer der Zukunft“ bezeichnet, muss geeignet in den Fachunterricht integriert werden. Die informatische Bildung hat die Aufgabe, die notwendigen Grundlagen zu vermitteln. Diesem Beitrag liegen Projekte zugrunde, die derzeit im Rahmen der „Intel/FSC AMILO-Mini –Schulprojekte 2009/2010“ mit zwei dritten Klassen durchgeführt werden. Die eigentliche Projektphase dauert sechs bis 10 Wochen. In der Vorbereitungsphase werden die Lehrerinnen und Lehrer mit der effizienten Nutzung der Technik vertraut gemacht und bei der Entwicklung von Einsatzkonzepten für den Fachunterricht unterstützt. Es ist eine Fortführung des Projektes in den vierten Klassen geplant. Die Ausstattung wurde durch ein interaktives Whiteboard ergänzt.

2 Computernutzung im Unterricht

Für die Primarstufe bzw. Grundschule gibt es keine bundeseinheitlichen Regelungen, welche Schuljahrgangsstufen dieser zuzuordnen sind. Die meisten Bundesländer ordnen die ersten 4 Schuljahre, Berlin und Brandenburg die ersten 6 Schuljahre der Primarstufe zu. Unsere Untersuchungen beziehen sich auf Sachsen-Anhalt. Hier werden in der Regel die Klassen 1 bis 4 als Primarstufe bezeichnet. Im folgenden Abschnitt wird die aktuelle Situation an Grundschulen bezogen auf die Lehrpläne von Sachsen-Anhalt betrachtet.

Im Jahre 2004 wurden die Bildungsstandards der Kultusministerkonferenz für den Primarbereich verabschiedet. Auf dieser Grundlage wurden für Sachsen-Anhalt neue Lehrpläne entwickelt und im Jahre 2007 eingeführt. Darin wurden auch Grundkompetenzen bezüglich der Computernutzung integriert. Schwerpunkt ist das Fach Deutsch. In diesem Fach wird eine Einführung in die elektronische Textverarbeitung gefordert. Weitere Angebote können von den Schulen im Rahmen ihrer ESA-Stunden (ESA – ergänzendes schulisches Angebot) schulspezifisch gemacht werden. Dieses Angebot wird an entsprechend ausgestatteten Schulen häufig genutzt, um den Schülerinnen und Schülern den Erwerb von Grundkompetenzen im Rahmen der computergestützten Präsentationsgestaltung zu ermöglichen.

Ergänzt werden die Lehrpläne durch ein kompetenzorientiertes Konzept „Medienbildung“ für die Grundschule, welches 2008 vom LISA Halle veröffentlicht wurde [LI08]. Dieses Konzept basiert auf dem Grundschullehrplan, in dem das Leben und Handeln in der Medienwelt als eine von vier Leitideen für die Erziehung und Bildung in der Grundschule betont wird. In diesem Konzept sind fünf Kompetenzbereiche definiert worden. Diese sind

1. Mit Informationen umgehen
2. Sich mit Hilfe von Medien austauschen
3. Medienprodukte erstellen
4. Medienangebote verstehen
5. Leben in der Medienwelt

Für die Umsetzung dieser Kompetenzbereiche ist kein eigenes Fach vorgesehen, sondern eine Integration in alle Fächer des Grundschullehrplanes. Es wird davon ausgegangen, dass die Schule über ein vernetztes Computerlabor bzw. Computerinseln mit Internetzugang verfügt. Die Schüler sollten einen persönlichen Account haben und über ein eigenes E-Mail-Konto verfügen.

Voraussetzung für die Nutzung des Computers ist der Erwerb von Grundkompetenzen in der Computernutzung. Im Grundschulbereich hat die „digitale Spaltung“ gegenüber der frühkindlichen Erziehung weiter zugenommen. Einige Schülerinnen und Schüler haben aus ihrem privaten Umfeld oder aus dem Bereich der Vorschulerziehung erste Kompetenzen im Umgang mit Computern erworben. Einige Schülerinnen und Schüler haben

noch keine Erfahrung im Umgang mit Computern gesammelt. Die Nutzung des „Klassenzimmers der Zukunft“ mit vollständig homogener Computerausstattung leistet einen wesentlichen Beitrag dazu, bei allen Schülerinnen und Schülern Grundkompetenzen zur Nutzung des Computers herauszubilden. Die im Rahmen des Projektes durchgeführten Untersuchungen zeigten, dass alle Schülerinnen und Schüler der 3. Klasse Erfahrungen bei der Arbeit mit Computern hatten. Nach einer Einweisung von ca. 30 Minuten in die Funktionsweise der Notebooks konnten alle Schülerinnen und Schüler mit den Computern arbeiten.

Die Schülerinnen und Schüler lernen die wichtigsten Systemkomponenten ihres Computerarbeitsplatzes kennen und informatisch korrekt zu benennen. Sie entwickeln Grundfertigkeiten bei der Bedienung der Eingabegeräte. Die klassischen Eingabegeräte sind Tastatur und Maus bzw. Touchpad. Üblicherweise sind diese ergonomisch an die Handgröße von Erwachsenen angepasst und daher für Kinder in der Grundschule eher ungeeignet. Es ist zweckmäßig, spezielle Computer und Eingabegeräte einzusetzen, auf die im folgenden Abschnitt noch ausführlicher eingegangen wird. Da bisher in den Bildungsstandards und Lehrplänen hauptsächlich von einer Nutzung des Computers im Deutschunterricht ausgegangen wird, ist die Tastatur für die nächste Zeit noch das dominierende Eingabegerät. Daher ist es sinnvoll, schon in der Grundschule einen Kurs bzw. ein Projekt im Maschinenschreiben anzubieten. Das setzt aber voraus, dass Tastaturen vorhanden sind, die an die Handgröße der Kinder angepasst ist.

Ein weiteres Eingabegerät ist das graphische Tablett bzw. der Tablet-PC. Die Stifteingabe bzw. die Touchscreen-Bedienung ermöglicht, den Computer als weiteres Zeichen- und Malwerkzeug zu nutzen und Texte handschriftlich zu erfassen. Der derzeitige Stand der Technik bietet diese Möglichkeiten als Ergänzung zum Erwerb der klassischen Kompetenzen im Schreiben, Zeichnen und Malen an. Der Lehrer hat damit die Möglichkeit, die Stifteingabe zum Methodenwechsel einzusetzen. Es können z.B. vorbereitete Arbeitsblätter mit Hilfe der Stifteingabe ausgefüllt werden. Für das Schreiben längerer Texte sind die Graphiktablets weniger geeignet, da die Schrift nicht mehr an der Stelle entsteht, an der geschrieben wird, sondern auf dem Bildschirm. Es erfordert einige Übung vom Nutzer, bis z.B. zügig ein Umlaut oder der Buchstabe i geschrieben werden kann, da zum Setzen der Punkte erst einmal die richtige Position auf dem Tablett gefunden werden muss. Der Einsatz eines Tablett-PCs erleichtert die Eingabe etwas, da direkt auf dem Bildschirm geschrieben wird. Schönschrift ist aber auch hier bei den derzeitigen verfügbaren Geräten nicht zu erwarten, da die Computer eine gewisse Höhe haben und eine ergonomische Handhaltung nicht möglich ist. Im durchgeführten Projekt wurden die graphischen Tablets etwa 3 Wochen später ausgegeben als die Notebooks. Bei den Schülerinnen und Schülern bestanden Anfangsschwierigkeiten im Umgang und die Euphorie, dieses Eingabemedium nutzen zu dürfen, war relativ gering. Nach einer Einarbeitungsphase mit Malprogrammen haben die Schülerinnen und Schüler das graphische Tablett als geeignetes Hilfsmittel zur Bearbeitung vorgegebener Arbeitsblätter im .pdf-Format eingesetzt.

Das Standardausgabegerät ist der Bildschirm. Bei der Nutzung des Computers im Unterricht sollte darauf geachtet werden, dass die Zeiträume, in denen ausschließlich am Bildschirm gearbeitet wird, nicht zu groß werden, um einer Ermüdung der Augenmuskulatur durch dauerhaftes Fokussieren auf einen bestimmten Arbeitsabstand vorzubeugen. Für viele Anwendungen, besonders im Sprachunterricht ist eine Audio-Ausgabe erforderlich. Einige Lernprogramme enthalten auch Audiodateien mit Erklärungen und Hilfestellungen zur Bearbeitung der Aufgaben. Hier ist es für den Unterrichtseinsatz auch aufgrund der Qualität der internen Lautsprecher zweckmäßig, Kopfhörer zu verwenden, auch um gegenseitige Störungen zu minimieren.

In dem der von uns begleiteten Schulprojekte sind die Schülercomputer mit dem Betriebssystem Microsoft Windows 7 ausgestattet. Als Anwendersoftware ist in der von uns verwendeten Basiskonfiguration ein Office-Paket enthalten. Ergänzt wird die die Softwareausstattung durch spezielle Lernsoftware. Zum Einsatz kommt die „Lernwerkstatt Grundschule“ Version 7 der Medienwerkstatt Mühlacker.

Im Rahmen des Konzeptes Medienbildung [LI08] werden von den Schülerinnen und Schülern Kompetenzen erwartet, die installierte Software starten, deren elementare Funktionen erklären und einsetzen zu können. Sie erlernen ihre Eingabegeräte und die graphische Benutzungsoberfläche zu bedienen.

Im Bereich der informatischen Grundkompetenzen sind die Schülerinnen und Schüler in der Lage, ihre Arbeitsergebnisse in Dateien abzulegen und zur Weiterbearbeitung wieder aufzufinden. Sie kennen die Grundbegriffe Datei und Ordner und sind in der Lage, eine geeignete Ordnerstruktur auf ihrem Computer anzulegen. Die Schülerinnen und Schüler können zwischen lokalen Anwendungen, der Arbeit in Schulnetz und der Arbeit im Internet unterscheiden.

Für den Kompetenzbereich „Mit Informationen umgehen“ bietet das 1:1-Konzept des persönlichen Computers wesentliche Vorteile. In der ersten Phase des Projektes stand für die Schülerinnen und Schüler kein einheitliches Mailsystem zur Verfügung. Es wurde ein Blog zum Informationsaustausch eingerichtet. Daran erlernten die Schülerinnen und Schüler wesentliche Grundlagen des Umgangs mit Informationen im Netz. In diesem Zusammenhang wurden z.B. folgende Fragen geklärt. Welche Informationen stelle ich mit welchem Ziel ins Netz? Wer kann die Informationen lesen? Wer darf im Blog schreiben bzw. Kommentare hinzufügen? Damit werden Kompetenzen erworben, die einen Beitrag zur Herausbildung eines kritischen Umgangs mit Informationen fördern.

Für den im Grundschullehrplan Sachsen-Anhalts vorgeschlagenen Einsatz im Deutschunterricht eignet sich besonders die Erstellung von Medienproduktionen. Der klassische Weg zur Herstellung ist die Gestaltung von Plakaten oder Wandzeitungen. Mit dem Computer ändert sich die Arbeitsweise der Schülerinnen und Schüler grundlegend. Bei der klassischen Gestaltung von Plakaten oder Wandzeitungen entwickeln die Schülerinnen und Schüler die Präsentation um das gefundene Material herum. Wird der Computer eingesetzt, so passen die Schülerinnen und Schüler das Material entsprechen ihrer Vorstellungen in das Konzept ein und experimentieren mit der Gestaltung. Klassische Informationsquellen, wie Bücher und Zeitungen werden durch Internetdienste ergänzt. Ein

wesentlicher Unterschied zeigt sich auch bei der Durchführung von Berichtigungen. Fehler in digital erstellten Präsentationen lassen sich wesentlich schneller berichtigen, als bei der klassischen Herangehensweise.

Der ständig verfügbare Computer im Grundschulbereich eröffnet auch die Möglichkeit, in jeder Unterrichtssituation bei Bedarf auf Informationen aus dem Internet zuzugreifen. Über geeignete und jugendschutzkonforme Zugänge erlernen die Schülerinnen und Schüler den verantwortungsbewussten Umgang mit dem Internet. Der Schwerpunkt der Arbeit ist Informationen finden, bewerten und präsentieren. Für das Finden von Informationen wurde in der ersten Projektwoche die Arbeit mit Suchmaschinen behandelt. Die Schülerinnen und Schüler lernen unterschiedliche Suchmaschinen kennen und erkennen das gleiche Anfragen unterschiedliche Suchergebnisse liefern. Schrittweise erlernen die Schülerinnen und Schüler, die Suchergebnisse zu vergleichen und zu bewerten. Bei der Arbeit in Gruppen werden die gefundenen Ergebnisse diskutiert und es bildet sich eine Bewertungskompetenz heraus.

Die Nutzung des Internets durch die Schülerinnen und Schüler erfordert eine altersspezifische Einführung in rechtskonforme Verhaltensweisen im Internet. Es werden die Grundlagen des Urheberrechtes und des Lizenzrechtes vermittelt. Besonders wichtig ist die Einhaltung der Urheberrechte bei Bildern und Musik z.B. bei der Erstellung multimedialer Präsentationen, besonders wenn diese anschließend auf dem Schulserver veröffentlicht werden sollen.

Ein weiteres Einsatzgebiet der Computer in der Grundschule ist der naturwissenschaftliche Unterricht. Der Computer kann für Aufgaben aus dem Bereich Messen-Steuern-Regeln eingesetzt werden. Ein Beispiel ist die Verwendung von Lego-Education-WeDo. Damit können einfache Maschinen konstruiert werden. Zur Steuerung der Motoren und zur Auswertung der Sensoren kann der Computer eingesetzt werden. Die notwendigen Algorithmen werden aus Grundbausteinen graphisch zusammengesetzt und parametrisiert.

Ein wesentlicher Vorteil des Einsatzes von Notebooks im Unterricht ist die Unterstützung der Differenzierung. Die Erfahrungen des Einsatzes der „Lernwerkstatt“ zeigen, dass jeder Schülerin und jedem Schüler mit vertretbarem Aufwand Aufgaben zur Verfügung gestellt werden können, die ihrem individuellen Leistungsniveau entsprechen. Leistungsstarke Schüler haben nicht mehr das Gefühl, mit zusätzlicher Arbeit belastet zu werden, sondern sie dürfen schon in der nächsten Leistungsstufe, aus Spielen als nächstes Level bekannt, arbeiten. Im begleiteten Projekt einer 3. Klasse wird das Notebook bis zu 80 % der Unterrichtszeit in den Fächern Deutsch, Mathematik und Sachkunde eingesetzt. Die Arbeitszeit liegt unterhalb der Akkulaufzeit der Computer, so dass ein Laden im Unterricht nicht notwendig ist.

Mit der Einführung des Klassenzimmers der Zukunft ergeben sich neue Anforderungen an die IT-Infrastruktur der Schule, die im folgenden Abschnitt näher betrachtet werden sollen.

3 Schulinfrastruktur

Die derzeit vorherrschende Computer-Infrastruktur an Grundschulen ist das Computerkabinett bzw. Computerinseln mit Standard-PCs. Bei günstigen Bedingungen teilen sich bei Nutzung des Kabinetts zwei Schülerinnen und Schüler einen Computerarbeitsplatz. Als Eingabegeräte stehen Tastaturen und Mäuse zur Verfügung, die für die Nutzung durch Kinderhände nur eingeschränkt geeignet sind. Die Computer sind vernetzt und an einen Schulserver mit Internetzugang angeschlossen. Für die Nutzung des Internets verfügt der Schulserver über einen geeigneten Kontentfilter. Die Nutzung des Computers erfordert den Wechsel in Computerkabinett und der Unterricht erfolgt in der Regel in der gesamten Zeit am Computerarbeitsplatz.

Eine Alternative zum Computerkabinett ist das mobile Klassenzimmer. Dieses wird von verschiedenen Herstellern seit einigen Jahren angeboten, hat sich jedoch bis heute nicht durchgesetzt. Als Computer kommen Notebooks oder Netbooks zum Einsatz, die in einem stabilen Rollwagen gelagert werden. Der Rollwagen ist gleichzeitig Ladestation für die Computer und kann zusätzlich noch einen Beamer, Drucker und Soundsystem aufnehmen. Zusätzlich sollte noch ein Lehrerarbeitsplatz integriert sein. Für die Kommunikation ist ein WLAN in allen Räumen erforderlich bzw. eine Netzwerkverkabelung mit einem Accesspoint auf dem Rollwagen.

Der Vorteil dieses Ansatzes liegt darin, dass mit dem Rollwagen die Technik für einen modernen Unterricht in jeden Klassenraum gebracht werden kann. Die Computer werden nur dann genutzt, wenn es für die Unterrichtssituation nutzbringend ist. Der Nachteil dieses Ansatzes besteht darin, dass der Wartungs- und Installationsaufwand der Computer von der Schule realisiert werden muss und keine persönlichen Daten auf den Geräten verbleiben können. Für die Erledigung von Hausaufgaben stehen diese Geräte auch nicht zur Verfügung. Damit ist dieser Ansatz nur als Übergang zur vollständigen Ausstattung aller Schülerinnen und Schüler mit persönlichen PCs zu sehen.

Bei der Einführung des „Klassenzimmers der Zukunft“ ergeben sich neue Anforderungen an die Schulinfrastruktur. Jede Schülerin und jeder Schüler erhalten ihr persönliches Notebook. Dieses ist sowohl in der Schule als auch zu Hause nutzbar. Zu beachten ist bei der Nutzung, dass dieser Computer vorrangig ein Lernwerkzeug ist. Der XO steht in Deutschland derzeit nicht zur Verfügung. Aus ergonomischen Gesichtspunkten eignen sich als Alternative Computer der „Netbook-Klasse“. Ein erster Vertreter dieser Klasse war der INTEL-Classmate. Dessen Nachfolgemodell ist wesentlich leistungsfähiger und wird mit einem Touchscreen ausgestattet sein. Damit ergeben sich für die Schülerinnen und Schüler neue Möglichkeiten der Interaktion.

Wichtig ist dabei, dass die gesamte Klasse mit einer homogenen Hardware und einer homogenen Basissoftware ausgestattet ist. Im Rahmen des von uns derzeit durchgeführten Projektes kommen die AMILO-Mini-Netbooks zum Einsatz. Zur Erweiterung der Eingabemöglichkeiten sind die Computer mit Wacom Bamboo-Graphiktablets ausgestattet. Jeder Lehrer sollte über ein leistungsfähiges Notebook, bezogen auf die Verwendung einer interaktiven Tafel, möglichst einen Tablet-PC, verfügen. Die im Projekt verwendete Hardware, Netbook und Graphiktablett wird von den Autoren als nicht optimal eingeschätzt, war aber im Rahmen der Initiative vorgegeben. Als geeigneter wird ein derzeit in der Entwicklung befindliches Notebook mit Tabletfunktion (convertible classmate PC) [CM09] beurteilt.



Abbildung 2: Persönlicher Schülerarbeitsplatz mit AMILO-Mini und Graphiktablett

Für das Klassenzimmer der Zukunft ist eine neue Form der Vernetzung der Schule zu realisieren. Die Schule muss über einen leistungsfähigen Schulserver verfügen. Dieser hat die Aufgabe zentrale Applikationen und Arbeitsmaterialien bereitzustellen. Es sollte eine zentrale Backup-Möglichkeit geschaffen werden um einen Schülernotebook im Falle eines Ausfalls schnell wieder herstellen zu können. Bei geeigneten Anwendungen ist kollaboratives Arbeiten zu ermöglichen. Der Schulserver ermöglicht als Internet-Gateway für alle Schülercomputer den Internetzugang. Gleichzeitig werden durch entsprechende Hard- bzw. Softwarelösungen die Anforderungen des Jugendschutzes bei der schulischen Internetnutzung realisiert.

Eine Voraussetzung, die Einsatzfähigkeit des „Klassenzimmers der Zukunft“ voll auszunutzen, ist das Vorhandensein einer multimedialen Schultafel. Dazu gehört ein interaktives Whiteboard in Verbindung mit multimedialer Software. Es gibt dabei unterschiedliche Konzepte. Einige Tafeln verhalten sich wie graphische Tablett, sind also nur mit aktiven Stiften bedienbar. Andere Modelle verhalten sich wie Touchscreens, deren Eingaben ebenfalls mit elektronischem Stift, aber auch mit den Fingern erfolgen können.

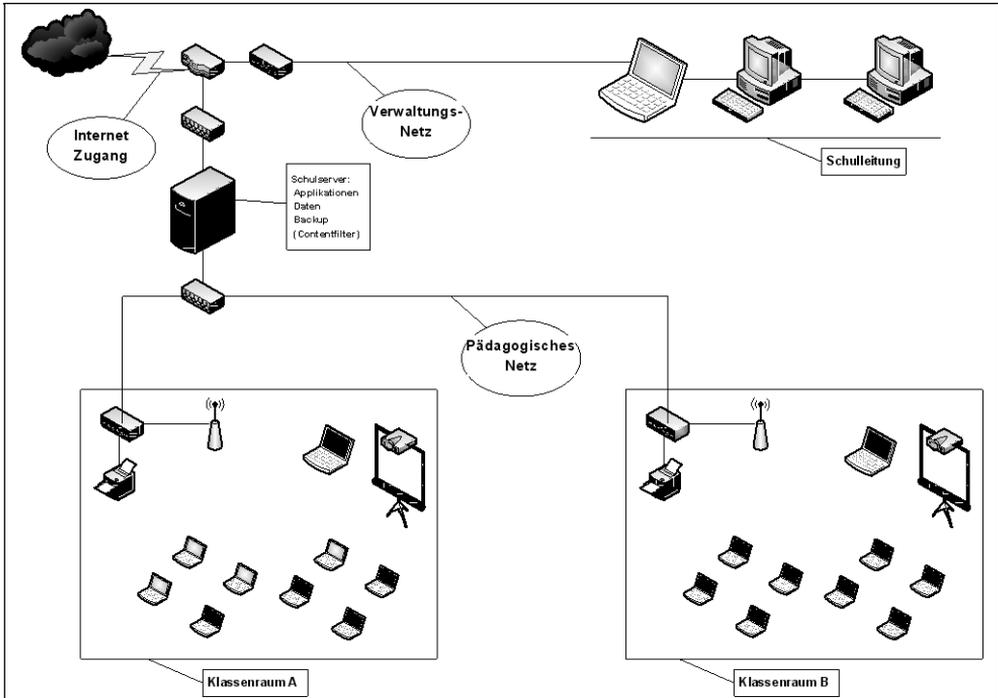


Abbildung 3: Schul-IT-Infrastruktur im Grundschulprojekt

In Verbindung mit einem pädagogischen Netzwerk für können Übertragungs- und Steuerungsfunktionen zwischen Lehrer- und Schülercomputer realisiert werden. Dazu gehört z.B. das Übertragen des Bildes von einem Schülercomputer auf den Projektor, die Übertragung des Bildes eines Schülercomputers auf den Lehrercomputer und in umgekehrter Richtung die Übertragung des Bildes des Lehrercomputers auf alle Schülercomputer. Weiterhin sollte die Möglichkeit bestehen, alle Bildschirme der Schülerarbeitsplätze dunkel zu tasten und deren Eingaben zu blockieren.

4 Schlussfolgerungen

Die wichtigste Grundlage für die Einführung des „Klassenzimmers der Zukunft“ im Bereich der Grundschule ist die Überzeugung aller Fachlehrerinnen und –lehrer, dass die Nutzung des Computers im Unterricht zu einem effizienteren Unterricht führt und die Motivation der Schülerinnen und Schüler verbessert. Der Computer darf nicht als zusätzliche Belastung empfunden werden, sondern muss als zusätzliche Chance erkannt werden. Dazu ist es in der ersten Phase erforderlich, die Lehrkräfte mit den neuen Werkzeugen vertraut zu machen und mit ihnen zu diskutieren, wie sie den Computer sinnvoll in den Unterricht der einzelnen Fächer integrieren wollen und welche Kompetenzen mit dem Computer besser erreicht werden können. Die Fachdidaktiken haben die Aufgabe, die notwendigen Grundlagen zu erarbeiten.

Da die Computer nicht mehr Bestandteil der Schulausstattung sind, sondern die Schule nur noch die Infrastruktur bereitstellt, um die Computer als Lernwerkzeug zu nutzen, müssen die Eltern wesentlich stärker in den Einführungsprozess einbezogen werden. Die Eltern müssen davon überzeugt sein, dass Computer einen Beitrag zur besseren Bildung der Kinder leisten. Da der Computer auch im Hausaufgabenbereich eingesetzt wird, sollten die Eltern in die Lage versetzt werden, die Tätigkeit der Kinder am Computer im häuslichen Bereich zu bewerten und zu steuern. Dazu ist auch eine Form der Weiterbildung für die Eltern notwendig. Eine Möglichkeit dazu bieten thematische Elternversammlungen.

Mit der Einführung des Computers in der Primarstufe wird ein wesentlicher Bestandteil zur Medienbildung geleistet. Die Schülerinnen und Schüler erlernen den verantwortungsbewussten kreativen Umgang mit dem Medium Computer. Damit dominieren die Möglichkeiten und Chancen eines ungehinderten Informationszugangs und einer ortsunabhängigen Kommunikation gegenüber den Gefahren und Risiken für die Kinder. Der Einsatz des Computers in der Grundschule ermöglicht es, einen wesentlichen Beitrag zu leisten, um frühzeitig individuelle Defizite bei Schülerinnen und Schülern auszugleichen.

Mit dem „Klassenzimmer der Zukunft“ wird sichergestellt, dass alle Schülerinnen und Schüler am Ende der Grundschule über Grundkompetenzen verfügen, den Computer und seine Software selbst bestimmt und zielgerichtet als Hilfsmittel zur Lösung von Problemen einsetzen zu können. Es wird ein wesentlicher Beitrag dazu geleistet, die Medienkompetenz zu fördern und ein aktives, selbst bestimmtes Leben in der Medienwelt zu führen.

„Nur durch eine derart frühe schulische Verankerung erster informatischer Inhalte kann sozialen und geschlechtsspezifischen Benachteiligungen vorgebeugt und damit die Chancengleichheit für alle Schülerinnen und Schüler gewahrt werden.“ [GI00]

Literaturverzeichnis

- [CM09] Classmate PC: www.classmatepc.com.
- [GI00] Gesellschaft für Informatik (GI) e.V.: Empfehlungen für ein Gesamtkonzept zur Informatischen Bildung an allgemein bildenden Schulen, Bonn, 2000
- [GI08] Gesellschaft für Informatik (GI) e.V.: Grundsätze und Standards für die Informatik in der Schule Bildungsstandards Informatik für die Sekundarstufe I, Bonn, 2008
- [KO06] Kochan, Barbara; Schröter, Elke: Abschlussbericht über die wissenschaftliche Projektbegleitung zur Bildungsinitiative von Microsoft Deutschland und Partnern – “Schlaumäuse – Kinder entdecken Sprache”. Berlin, 2006. www.schlaumaeuse.de/Informationen/Mediathek/Abschlussbericht_final.pdf
- [LI08] Bartsch, P. D. et.al.: Ein kompetenzorientiertes Konzept für die Grundschule mit Beispielaufgaben und einem Medienpass. Landesinstitut für Lehrerfortbildung, Lehrerweiterbildung und Unterrichtsforschung von Sachsen-Anhalt (LISA), Halle, 2008
- [MI07] Mitzlaff, Hartmut: Computer (ICT), Grundschule, Kindergarten und neue Lernkultur. Schneider Verlag Hohengehren, Baltmannsweiler, 2007.
- [NE05] Negroponte, N.: One Laptop Per Child (OLPC). www.laptop.org.

Lernen mit Etoys

Rita Freudenberg

FIN-ISG
Otto-von-Guericke-Universität
PSF 4120
39016 Magdeburg
rita@isg.cs.uni-magdeburg.de

Abstract: In diesem Vortrag wird der Einsatz von Squeak-Etoys im Bereich der Primarstufe vorgestellt. In verschiedenen Ländern gibt es Pilotprojekte, die sich mit dem Einsatz von Computern in dieser Alterstufe befassen. Durch die Verbreitung der XO-Laptops wird in vielen Ländern mit Etoys gearbeitet. In diesem Beitrag wird untersucht, wie sich Etoys in die Rahmenbedingungen und Lehrpläne der jeweiligen Länder integrieren lässt und wie sich das auf Deutschland übertragen lässt.

1 Was ist Etoys?

Die Entwicklung von Etoys wurde inspiriert durch Programme wie LOGO, Hypercard, PARC-Smalltalk und starLOGO und die pädagogischen Ideen von Maria Montessori, Seymour Papert und Jerome Piaget.

Die Grundidee von Etoys besteht darin, eine einfache, aber mächtige Benutzungsoberfläche zu schaffen, die „überall und auf allem“ funktioniert und den Endnutzern die Möglichkeit gibt „auf alles zuzugreifen und alles zu verändern“. Alle Objekte können überall erstellt und überall verwendet werden, Nutzer können mit den vorhandenen Objekten arbeiten oder ebensolche Objekte selbst erstellen. Im Gegensatz zur Idee der „Applikation“ (die aus den 60ern stammt) die die Nutzer jeweils auf die Möglichkeiten dieser einen Applikation beschränkt, ist Etoys wie eine „Meta-Applikation“ zu verstehen, eine Sammlung aller schon erstellten Dinge, in der jedes Objekt gleichzeitig erstellt, manipuliert und dargestellt wird.

Dabei steht das, was die Nutzer erschaffen, im Mittelpunkt, und nicht eine Vielzahl von Auswahlmöglichkeiten und Schaltknöpfen zur Verwendung des Systems. Deshalb ist auf dem Startbildschirm nicht viel zu sehen von den Ressourcen, die Etoys zur Verfügung stellt. Die Nutzer erstellen Dinge, von einfachen Präsentationen bis zu komplexen Simulationen und können dafür über nahezu den gesamten Bildschirm verfügen. Vorhandene Ressourcen befinden sich in versteckten Fenstern, sogenannten „Klappen“.

Nutzer können beliebig viele Klappen für eigene Projekte erstellen, die entweder nur zu einem Projekt gehören, aber auch an andere Projekte weitergegeben werden können. Zu den vorhandenen Objekten gehören verschiedene graphische Formen, Audio- und Videoplayer, Spiele und ein System zur Partikelsimulation. Ein Projekt befindet sich auf einer Bildschirmseite, aber ein Projekt kann aus mehreren Seiten bestehen. Ähnlich einer Powerpoint-Präsentation hat man eine Menge von Seiten, die von Hand oder mit einem Skript sortiert werden können, aber komplexere Inhalte haben können.

Die Etoys-Benutzeroberfläche verwendet nicht die in vielen Anwendungen üblichen „Modi“ wie Editier-Modus, Präsentationsmodus usw., sondern erlaubt jederzeit die Manipulation aller Objekte. Diese erfolgt über direkt am Objekt angeordnete kreisförmige Symbole, sogenannte „Halos“, die u. a. Rotation, Kopieren, Verzerren und Löschen des Objektes erlauben. Dieses Menü gibt es für jedes visuell sichtbare Objekt auf dem Bildschirm, für vom Nutzer gemalte Objekte ebenso wie für Skripte und selbst für jedes der Halo-Symbole. Das erlaubt es dem Nutzer, das gesamte System mit allen seinen Bestandteilen individuell anzupassen. Die Ähnlichkeit der Objekte in Etoys hört aber nicht mit den Halos auf, sie geht tiefer bis in die Implementierung der Objekte selbst. Die visuelle Unterscheidung ist nur ein „Kostüm“, das jedes Objekt trägt (vergleichbar einem Schauspieler, der verschiedene Rollen spielt).

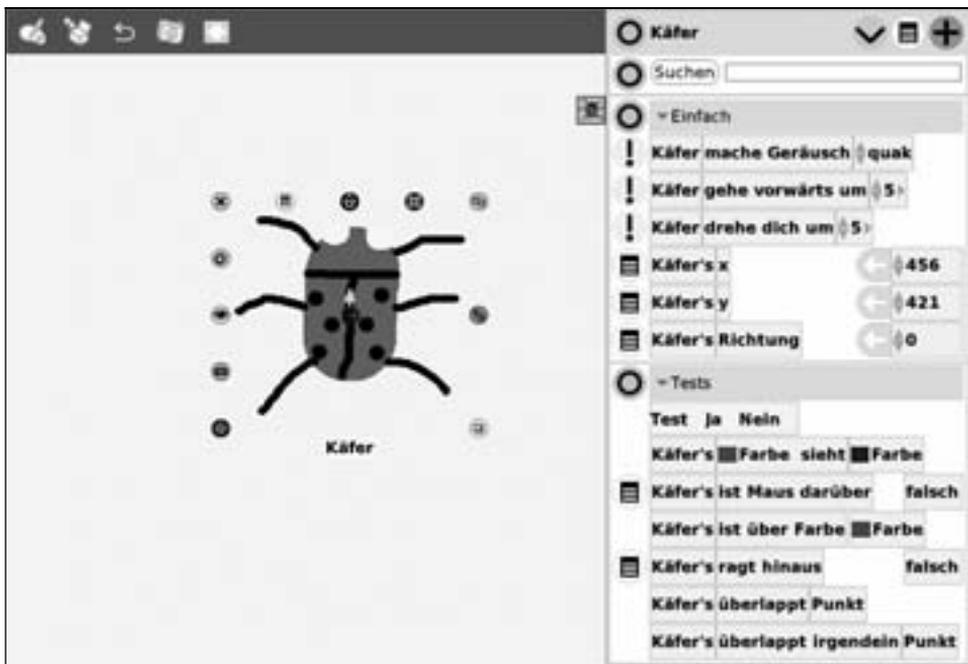


Abbildung 1: Zwei Kostüme für das Objekt „Käfer“

Das türkisfarbene Halo-Symbol ist besonders wichtig, denn es öffnet einen Betrachter für ein Objekt, der Eigenschaften und Verhalten des Objektes in Kategorien sortiert anzeigt. Für alle vom Benutzer erstellten Objekte sind diese Kategorien gleich: das Objekt hat eine graphische Darstellung, es kann andere Objekte beinhalten, es kann Skripte erhalten, es hat einen Stift usw. Der Betrachter eines Objektes ist aber nichts anderes als ein anderes Kostüm für das Objekt! Und der Nutzer kann beliebig viele weitere Kostüme erstellen.

Die Skripte für Objekte werden immer nach demselben Prinzip erstellt: indem Kacheln aus dem Betrachter eines Objektes in ein Skriptfenster gezogen werden. Der Grund für die Verwendung von Kacheln ist die Vermeidung von Syntaxfehlern, so dass sich Nutzer auf die Problemlösung konzentrieren können. Durch die Übersetzung in verschiedene Sprachen sind die meisten Kacheln selbsterklärend.

Mit Etoys ist alles konstruierbar, alles, was vorhanden ist, ist veränderbar und alles kann bis in seine Details untersucht und „auseinander genommen“ werden. Diese Philosophie trifft nicht nur für Etoys zu, sondern zieht sich bis in das darunter liegende Squeak Smalltalk System, was versteckt, aber nicht versperrt ist und so eine „white box“ darstellt, die dem Nutzer die Möglichkeit gibt, das gesamte System zu kontrollieren (vgl. [AK]).

2 Etoys im Schuleinsatz

Etoys war von Anfang an auf allen XO-Laptops enthalten, die ausgeliefert wurden. Für viele Lehrer war es das Programm mit den vielversprechendsten Einsatzmöglichkeiten im Unterricht und so wird es in vielen Schulen weltweit eingesetzt. Für die Kinder in den OLPC-Projekten ist der XO-Laptop der erste Kontakt mit einem Computer überhaupt und sie brauchen Zeit, um die grundlegenden Fertigkeiten im Umgang damit zu erlernen. Die motorischen Fertigkeiten sind durch regelmäßige Benutzung recht schnell erlernt, schwieriger ist es, die der Benutzung zugrunde liegenden Prinzipien zu verstehen. Das erleichtert es aber auch, neue Metaphern für die Benutzung zu verwenden, da nicht umgelernt werden muss. Das trifft auch auf die nicht den Standardprogrammen entsprechende Benutzungsoberfläche von Etoys zu.

2.1 Deutsch – Geschichten erzählen mit Animationen

Geschichten zu erzählen hat eine lange Tradition in der Menschheitsgeschichte. Untersuchungen haben gezeigt, dass sich Fakten leichter erinnern lassen, wenn sie im Kontext einer Geschichte vermittelt wurden (siehe [MT]).

Die derzeitige gesellschaftliche Entwicklung zeigt, dass immer mehr Prozesse automatisiert und von Computern ausgeführt werden können. Das Finden von Fakten beispielsweise geht über Suchmaschinen und Datenbanken um Größenordnungen schneller, als es früher mit Büchern und dem menschlichen Erinnerungsvermögen möglich war. Was Computer aber nicht so einfach können, ist, diese Fakten in einem

Kontext zu sehen und miteinander in Verbindung zu bringen. Diese Fähigkeit hat der Mensch dem Computer voraus, auch deshalb wird sie in Zukunft immer wichtiger werden und sollte in der Schule erlernt werden. Und Fakten im Kontext sind Geschichten (vgl. [DP]).

In den meisten Fällen beginnt die Einführung von Etoys mit der Verwendung des Malwerkzeuges. Die Kinder können kreativ sein und entweder nach vorgegebenen Themen oder eigenem Interesse Bilder erstellen. In Etoys entstehen allerdings nicht einfach Bilder, sondern Objekte. Diese Objekte lassen sich manipulieren, programmieren, man kann mit ihnen interagieren und sie zu Geschichten zusammenfügen. Da die Kinder die Animationen komplett selbst erstellen, lernen sie auch viele Ideen kennen, die sich hinter dem Animationsprozess verbergen. Das Prinzip einer Animation besteht darin, dass Einzelbilder schnell nacheinander angezeigt werden und so den Eindruck einer kontinuierlichen Bewegung vermitteln. Neben dem dafür erforderlichen Zeichnen der einzelnen Bilder erstellen die Kinder in Etoys auch das Skript für die abwechselnde Anzeige der Bilder selbst.

Auch Filme und Videos basieren auf diesem Grundprinzip (was man in Etoys sehen kann, wenn man anstatt selbst gemalter Bilder Videobilder verwendet), natürlich braucht man für einen Film wesentlich mehr Bilder, auch sind diese wesentlich größer und befinden sich normalerweise in einer Datei auf der Festplatte des Rechners. Und was passiert, wenn man Töne aufnimmt und abspielt? Wie funktioniert eine Präsentation? So lassen sich ausgehend von der Idee der Animation viele verschiedene Gebiete erforschen.

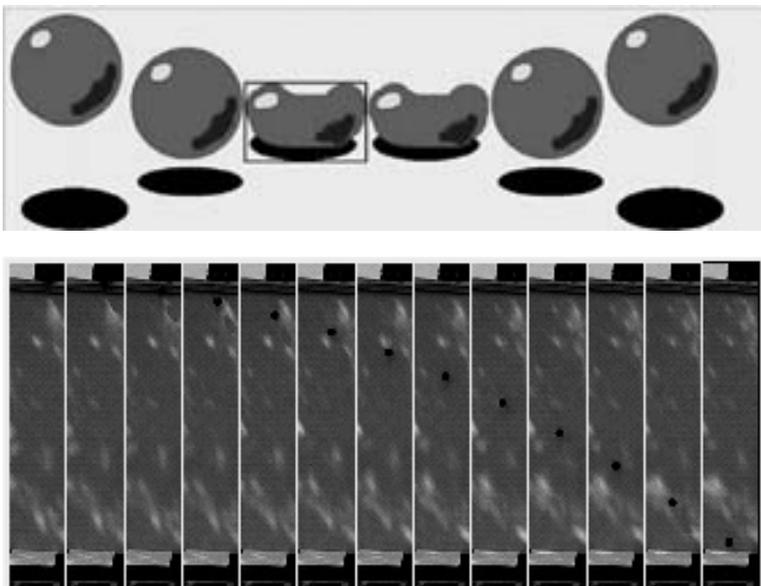


Abbildung 2: Gemalte Zeichnungen(oben) und Video Frames (unten) für eine Animation

In einer indischen Grundschule begannen Kinder damit, Bilder aus einfachen Formen und Farben zu zeichnen. Später wurden die Bilder komplexer, gehörten aber noch zu einer Zeichnung. Die Kinder fanden heraus, dass es einfacher ist, mehrere kleine Objekte zu zeichnen und diese dann zu einer größeren Szene zusammenzufassen. Nach ungefähr drei Monaten waren einige der Kinder in der Lage, kleine Animationen in ihre Szenen zu integrieren. Das heißt, sie können kleine Skripte für ihre gezeichneten Objekte erstellen. Diese Zeichnungen und Animationen sind nicht allein kreative Beschäftigungen. Ein Junge begann mit den fertigen Objekten „Stern“ und „Ellipse“ und lernte schnell, die Sterne sich um die Ellipse drehen zu lassen. Dabei kam er auf die Idee, das gleiche Skript zu verwenden, um einen Satelliten zu machen, der um die sich drehende Erde kreist. Aus einem Buch suchte er sich ein Bild des Aryabhata Satelliten heraus und die Erde zeichnete er nach dem Globus, der in der Schulbibliothek steht [SUB].



Abbildung 3: Indischer Junge vor seinem Projekt

Das Beispiel zeigt, dass die selbstständige Beschäftigung mit einem Thema in Etoys (Zeichnen und erste Animationen) das Lernen in vielen Gebieten motivieren kann. Etoys selbst schränkt die Kinder inhaltlich nicht ein.

Die Kinder der Guy Benjamin Grundschule in St. John auf den Virgin Islands erhielten im Januar 2008 die XO Laptops. In einem Pilotprojekt wurde die Einführung von Etoys von Januar bis März begleitet. Die Kinder erstellten im Laufe der drei Monate Geschichten über ihre Insel, die sie am Ende vor der Klasse präsentierten. Die Einführung erfolgte durch eine Lehrerin für alle Kinder, später halfen sich die Kinder gegenseitig weiter. Etoys bietet sehr viele Möglichkeiten, Ideen umzusetzen, die beim Experimentieren von den Kindern entdeckt werden, so dass es immer Dinge gibt, die einige Kinder wissen und andere nicht [WAV].

2.2 Mathematik – geometrische Formen und Zusammenhänge verstehen

Die im vorigen Kapitel gezeigten Ideen lassen sich für verschiedene Unterrichtsinhalte umsetzen, neben Sachkunde oder Deutschunterricht können auch im Mathematikunterricht Geschichten verwendet werden. Ein Beispiel dazu beschreibt Yasmin Kafai in ihrem Buch „Minds in Play“ [YK]. In einer Studie erhielten Kinder einer vierten Klasse den Auftrag, auf dem Computer ein Spiel zu entwickeln, mit dem Bruchrechnung erlernt werden kann. Die meisten Kinder erfanden eine Rahmenhandlung, in die sie Aufgaben integrierten. Die Studie wurde mit LOGO durchgeführt, Etoys verfügt verglichen damit über eine noch größere Bandbreite an Ausdrucksmöglichkeiten. An der University of Illinois at Urbana-Champaign hat Kathleen Harness Materialien für die Verwendung von Etoys im Grundschulbereich entwickelt [KH]. Im Einstiegsprojekt werden vorhandene Etoys-Objekte verwendet, die aus dem Lager entnommen und auf dem Bildschirm angeordnet werden. Damit kann mathematisches Vokabular für Position, Ort und räumlichen Beziehungen zwischen Objekten aufgebaut werden (über, unter, zwischen usw.). Die Kinder positionieren die Objekte auf dem Bildschirm und beschreiben dann für die Mitschüler, wo sie sich befinden, indem sie die erlernten Begriffe verwenden.

Für ältere Schüler können weitere unterschiedliche Objekte verwendet werden, die dann gruppiert oder nach Mustern angeordnet werden sollen. Damit lassen sich erste Vorstellungen von Brüchen veranschaulichen, z.B. durch Erkenntnisse wie „die Hälfte meiner Objekte sind Sterne“ oder „ein Viertel meines Bildschirms ist von Objekten bedeckt“. So sind erste Übungen mit Etoys möglich, die sehr geringe Einarbeitungszeit benötigen. Schritt für Schritt werden die Kinder mit den weiteren Möglichkeiten vertraut gemacht wie dem Malwerkzeug und dem Betrachter. Mit jedem Schritt können weitere Herausforderungen für die Kinder verbunden werden, so dass die Einführungsphase in die Bedienung von Etoys immer auch Kompetenzen vermittelt, die den Vorgaben des Lehrplanes entsprechen.

Aufgabe	Etoys-Bedienung	Kompetenz
Die Kinder bewegen ein Objekt über den Bildschirm und platzieren es an vorgegebenen Positionen.	Starten des Programms, Verwenden des Lagers, Drag&Drop	Verständnis von Formulierungen wie „Bewege das Objekt in die obere linke Ecke“, eigenes Formulieren solcher Anweisungen
Die Kinder schreiben Skripte, um Objekte über den Bildschirm zu bewegen	Verwendung des Halo-Menüs, Öffnen des Betrachters, Schreiben von Skripten, Kacheln für Bewegung und Richtung	Verstehen von Maßeinheiten, Erkennen sinnvoller Werte für Maße, Konzept des Koordinatensystems

Erstellen einer Zahlentafel als Vorratsbehälter für unendlich viele Zahlen und Zeichen zum Zählen, Anordnen und Rechnen, Erstellen magischer Quadrate	Verwenden von Malwerkzeug, Spielwiese und Behälter	Zahlenverständnis, Ordnen von Zahlen, Zählen, Zahlenreihen,
Zeichnen geometrischer Figuren mit dem eingebauten Stift der Objekte	Verwenden der Kategorie „Stifte“	Verständnis für den Zusammenhang zwischen Winkelgröße und Eckpunkte regelmäßiger Vielecke

Tabelle 1: Aufgabenstellungen für Grundschüler

Dies ist nur ein kleiner Ausschnitt der Aufgaben, die für den Einsatz in Illinois ausgearbeitet wurden. Die vermittelten Kompetenzen finden sich auch in vielen anderen Lehrplänen und lassen sich deshalb auch auf hiesige Grundschulen übertragen.

2.3 Etoys und informatische Bildung

In diesem Beitrag wird der Fokus auf den Einsatz von Etoys im Grundschulbereich gelegt. Ein Grund dafür ist die große Verbreitung der XO-Laptops und damit neuer Etoys-Nutzer und -Initiativen in dieser Altersstufe, ein weiterer, dass Schüler im Primarbereich heute in der Mehrheit Medienerfahrung haben, die in der Schule aufgegriffen, hinterfragt und weiterentwickelt werden sollte (vgl. Vortrag H. Herper). Die Verwendungsmöglichkeiten von Etoys sind aber nicht auf diese Altersstufe beschränkt.

Für die 10. Klasse gibt es Unterrichtsmaterial zur Einführung in die objektorientierte Modellierung [MS], in der Etoys für eine schnelle Modellentwicklung genutzt wird. An der Universität Magdeburg wird Etoys in Lehrveranstaltungen für die Lehramtsstudiengänge u. a. als interaktives Präsentationswerkzeug genutzt. Da von der Kacheldarstellung der Skripte auf den Smalltalk-Quelltext umgeschaltet und dann in Squeak weiter programmiert werden kann, ist ein Übergang zur Programmierung innerhalb der Umgebung möglich (siehe Abb. 4).

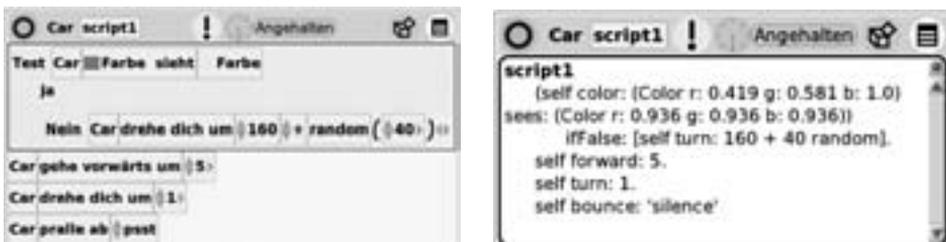


Abbildung 4: Ein Etoys-Skript in Kachel- und Quelltextdarstellung

Natürlich muss auch der Umgang mit Etoys wie mit jedem neuen Werkzeug erlernt werden. Die Erfahrungen zeigen, dass das Basiswissen in etwa 10 Lektionen zu je einer Stunde vermittelt werden kann. Wichtig für diese Lektionen ist, dass sie nicht nur dem Zweck dienen, die Bedienung und die Grundlagen von Etoys zu üben, sondern jeweils nur ein kleiner Aspekt besprochen und dann mit einer Aufgabenstellung verknüpft wird. So sind diese Lektionen nicht nur zusätzlich erforderliche Zeit, die ein Lehrer im Unterricht unterbringen muss, sondern vermitteln auch Lerninhalte, die ohnehin im Lehrplan stehen. Wird Etoys im Primarbereich erlernt und benutzt, kann es im Sekundarbereich mit weniger Einarbeitungszeit eingesetzt werden. Die Simulation physikalischer Vorgänge, die Visualisierung mathematischer Zusammenhänge oder die Gestaltung von Graphiken sind nur wenige Beispiele, wie Etoys in verschiedenen Fächern genutzt werden kann.

Zur informatischen Bildung im Primarbereich hat das Landesinstitut für Lehrerfortbildung, Lehrerweiterbildung und Unterrichtsforschung von Sachsen-Anhalt (LISA) ein Medienkonzept für die „Medienbildung in der Grundschule“ herausgegeben [LISA]. Darin werden fünf Kompetenzbereiche aufgeführt, die Relevanz für den Unterricht besitzen. Das sind:

- mit Informationen umgehen
- sich mit Hilfe von Medien austauschen
- Medienprodukte herstellen
- Medienangebote verstehen
- Leben in der Medienwelt

Diese Bereiche überlappen teilweise stark, der Bereich „Leben in der Medienwelt“ umfasst alle anderen Bereiche und geht noch darüber hinaus. Das Medienkonzept untersetzt diese Bereiche mit Leitideen, Kompetenzerwartungen, beispielhaften Inhalten und dem erforderlichen Grundwissen. Als Nachweis der erworbenen Medienkompetenz wird ein Medienpass vorgeschlagen.

Ähnliche Konzepte für den Kindergarten- und Grundschulbereich gibt es auch in anderen Bundesländern (siehe z. B. [TK], [RP]). In all diesen Konzepten wird kein neues Fach vorgeschlagen, sondern die Mediennutzung in den vorhandenen Fächerkanon integriert. Das ist auch erforderlich, um die komplexen Zusammenhänge erfahrbar zu machen und neben der Medienkompetenz auch Sach-, Methoden-, Sozial- und Selbstkompetenz bei den Schülerinnen und Schülern zu entwickeln.

Auch im OLPC-Projekt wurde schnell deutlich, dass eine Einführung der Laptops in Schulen ein Konzept braucht und die Lehrer entsprechend weitergebildet werden müssen. Diese entstanden entweder im Vorfeld oder begleitend zur Ausgabe der Laptops und werden ständig weiterentwickelt. Vorbildlich funktionierte dieser Prozess in Nepal, wo über ein Jahr vor dem Unterrichtseinsatz in Zusammenarbeit mit dem Kultusministerium, Lehrern und Softwareentwicklern Aktivitäten für die Rechner erstellt

wurden, die genau auf die Unterrichtsinhalte abgestimmt sind. In Uruguay wird jeder Grundschüler bis Ende dieses Jahres einen XO-Laptop haben. Aus den bisherigen Pilotprojekten dort wird deutlich, dass schon jetzt ein positiver Einfluss auf die Kinder und ihre schulischen Leistungen festzustellen ist. In einer ersten Studie wird aber auch klar, dass unbedingt in die Entwicklung der Software und der Lerninhalte investiert werden muss [HBCF].

Diese Erkenntnis kommt nicht überraschend und trifft für alle Länder zu, die Laptops in Schulen einsetzen wollen. Auch die Bedingungen für den Einsatz, also die erforderliche Schulung der Lehrkräfte und die Integration in bestehende Lehrpläne und den Schulalltag, unterscheiden sich nicht grundlegend von den Bedingungen hierzulande. Vergleichbar sind auch die Medienkompetenzen und Basisinhalte in den Fächern. Unterschiede gibt es vor allem in den Inhalten, mit denen die Kompetenzen vermittelt werden, wobei von Fall zu Fall entschieden werden müsste, ob diese länderspezifisch sind oder nicht. In einem südafrikanischen Projekt wurden beispielsweise comic-artige Anleitungen für Etoys entwickelt, die sicher auch deutsche Grundschüler ansprechen würden (siehe Abb. 5).



Abbildung 5: Eine Seite aus den Materialien des Kusasa-Projekts

Bei aller kulturellen Differenz heißt das, dass die Entwicklung von Schulungsmaterial länderübergreifend erfolgen kann und vorhandenes Material nach einer Übersetzung auch in anderen Ländern eingesetzt werden kann. So entwickelt die Squeakland Foundation derzeit einen beispielhaften Lehrplan zum Einsatz von Etoys in der Klassenstufe 4, der aus einer Basislektion zum Erlernen von Etoys und darauf aufbauenden Lektionen zu verschiedenen Themen besteht. Dieser Lehrplan wird begleitend in Grundschulen in den USA und Brasilien erprobt.

Für den Einsatz im Unterricht gibt es verschiedene Szenarien, beispielhaft seien hier genannt

- Die Schülerinnen und Schüler erstellen selbstständig Projekte nach von den Lehrern vorgegebenen Aufgabenstellungen. Hier kommt der konstruktivistische Ansatz zur vollen Entfaltung, da bei der Projekterstellung eine individuelle, ausführliche Beschäftigung mit dem Aufgabengegenstand erfolgt und das erarbeitete Wissen selbstständig transformiert wird.
- Die Schülerinnen und Schüler erhalten ein vorbereitetes Projekt, mit dem sie experimentieren können. Es können verschiedene Aufgaben gelöst werden. Je nach Aufgabe muss das Projekt genauer untersucht und gegebenenfalls verändert werden. Durch die Vorbereitung ist der Zeitaufwand beim Schuleinsatz weniger hoch, es kann ähnlich wie bei anderen Experimenten gearbeitet werden.
- Die Schülerinnen und Schüler verwenden ein vorbereitetes Projekt, um eine genau definierte Aufgabe zu lösen. Ein Beispiel hierfür wäre ein Vokabeltrainer, in dem Vokabeln und Bilder zugeordnet werden müssen (Beispiel aus Nepal). Hier soll das Projekt vom Schüler nur benutzt, aber nicht weiter bearbeitet werden. Geschulte Lehrkräfte könnten ein solches Projekt bearbeiten und z.B. die Bilder und Vokabeln an ihre Bedürfnisse anpassen.

Diese drei Varianten können in beliebiger Form gemischt auftreten. Sie beziehen sich auf den Einsatz von Etoys ohne Berücksichtigung des zugrundeliegenden Smalltalk-Systems.

3 Etoys und die Bildungsstands Informatik

Die Bildungsstandards Informatik [AKBSI] setzen bei der Jahrgangsstufe 5 an, das ist in den meisten Bundesländern nicht mehr der Primarbereich. Es wäre sinnvoll, die Inhalte, die für die Jahrgangsstufen 5 bis 7 formuliert sind, mit dem Blick auf den Medieneinsatz in der Grundschule zu untersuchen. Wird Etoys eingesetzt, heißt das, dass die Schüler Skripte schreiben, also programmieren. Sie bearbeiten damit Inhalte der Bereiche „Algorithmen“ und „Information und Daten“ der Bildungsstandards. Außerdem beschäftigen sie sich mit dem Prozessbereich „Modellieren und Implementieren“.

Untersucht man eine konkrete Aufgabenstellung, können auch andere Inhalts- und Prozessbereiche berührt werden. In der Grundschule steht meiner Meinung nach die Terminologie nicht im Vordergrund, so dass es weder offensichtlich ist noch vom Lehrer ausdrücklich erklärt werden muss, dass die Schüler gerade z. B. „Modellieren“. In der Sekundarstufe kann dann auf die Erfahrungen Bezug genommen und ein Begriff wie „Modellierung“ eingeführt werden.

Durch seine vielfältigen Einsatzmöglichkeiten können für viele der Inhalts- und Prozessbereiche Aufgabenstellungen gefunden werden, die mit Etoys realisierbar sind. Es wird interessant sein, zu verfolgen, inwieweit sich die Unterrichtsmaterialien, die in den verschiedenen Ländern entwickelt werden, in die Informatikstandards einordnen lassen.

Literaturverzeichnis

- [AK] Alan Kay, Squeak Etoys Authoring & Media, <http://www.squeakland.org>
- [AKBSI] Arbeitskreis „Bildungsstandards“ der Gesellschaft für Informatik (Hrsg.): Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I. Empfehlungen der Gesellschaft für Informatik e. V. vom 24. Januar 2008. In: LOG IN, 28. Jg. (2008), Heft 150/151, Beilage.
- [CR] B. J. Conn, Kim Rose, Powerful Ideas in the classroom.
- [DP] Daniel H. Pink, A whole new mind, Riverhead Books 2006.
- [HBCF] Juan P. Hourcade, Daiana Beitler, Fernando Cormenzana, Pablo Flores, Reflections on a Pilot OLPC Experience in Uruguay, Ceibal workshop, <http://www.cs.uiowa.edu/~hourcade/ceibal-workshop.pdf>
- [KH] Kathleen Harness, K5 Technology Passport Etoys, The Office for Mathematics, Science, and Technology Education, <http://www.etoysillinois.org/>
- [LISA] Landesinstitut für Lehrerfortbildung, Lehrerweiterbildung und Unterrichtsforschung (Hrsg.): Medienbildung in der Grundschule. LISA Halle, 2008.
- [MPFS] Medienpädagogischer Forschungsverbund Südwest (Hrsg.): KIM-Studie 2008, Kinder + Medien, Computer + Internet, Basisuntersuchung zum Medienumgang 6- bis 13-Jähriger in Deutschland, <http://www.mpfs.de/fileadmin/KIM-pdf08/KIM08.pdf>
- [MS] Markus Schlager, Objektorientierung und Modellierung, LSH Marquartstein, 2008, <http://signalbscw.tcs.informatik.uni-muenchen.de/pub/bscw.cgi/0/211260>
- [MT] Mark Turner, The Literary Mind: The Origin of Thought and Language, Oxford Press, 1996.
- [RP] Medienbildung in der Grundschule auf dem Bildungsserver Rheinland-Pfalz, <http://medienbildung-gs.bildung-rp.de/>
- [SUB] K. K. Subramaniam, Sikshana, <http://sikshana.blogspot.com/>
- [TK] Thüringer Kultusministerium (Hrsg.): Medienkompetenz in der Grundschule, Erfurt, 2004.
- [WAV] Waveplace Foundation, <http://waveplace.com/>
- [YK] Yasmin Kafai, Minds in Play, Computer Game Design as a Context for Children's Learning, LEA 1995.

Algorithmik in der Grundschule

Michael Weigend

Institut für Didaktik der Mathematik und der Informatik
Westfälische Wilhelms-Universität Münster
Fliednerstraße 21
Postleitzahl Münster
michael.weigend@uni-muenster.de

Abstract: Der Betrag präsentiert Ergebnisse einer Studie mit mehr als 120 Grundschulkindern zur Rezeption und Umsetzung informaler algorithmischer Texte. Das Setting besteht aus einer Befragung und zwei Spielen, bei denen die Kinder Bilder nach Anweisungen zeichnen. Die Spiele sind nicht nur ein Forschungsinstrument, sondern gleichzeitig auch Prototypen für grundschultaugliche Unterrichtsaktivitäten im Zusammenhang mit Sinn entnehmendem Lesen. Die gewonnenen empirischen Ergebnisse dienen als Grundlage für das Design multimedialer Lernspiele für Grundschüler zur Förderung algorithmischen Denkens.

1 Einführung

Ein Algorithmus ist eine präzise Beschreibung von Aktivität. Algorithmen sind die Grundlage jedes Computerprogramms und – allgemein – jedes Systems mit vorhersehbarem Verhalten. Auch natürliche und soziale Phänomene haben ihre algorithmische Seite. Normen, die das Zusammenleben in sozialen System regeln, sind zwar – für sich genommen – nicht immer als Algorithmen formuliert, führen aber während der Anwendung in einer konkreten Lebenssituation, zu verhaltenssteuernden Algorithmen („Wenn die Ampel rot ist, halte ich an.“ oder „Ich warte solange, bis alle mit dem Essen fertig sind.“). Die Fähigkeit Algorithmen zu verstehen, auszuführen und selbst zu entwickeln ist eine Facette von „computational thinking“ [W06]. Sie ist eine Basiskompetenz, die jedes Mitglied einer modernen Gesellschaft beherrschen sollte. In den Bildungsstandards Informatik der GI wird von Schülerinnen und Schülern in den Jahrgangsstufen 5 bis 7 die Fähigkeit erwartet, Algorithmen zu lesen und zu interpretieren [GI08, S. 30ff.].

In diesem Beitrag widme ich mich zunächst der Frage, in wieweit bereits Grundschüler (dritte und vierte Klasse) sich mit algorithmischen Texten auseinandergesetzt haben. In den anschließenden Abschnitten werden empirische Untersuchungen zur algorithmischen Leistungsfähigkeit von Grundschulern vorgestellt. Ich beschränke mich auf zwei Bereiche: Steuerung und Benennung. Der Begriff Steuerung bezieht sich darauf, in welcher Reihenfolge und unter welchen Bedingungen Instruktionen ausgeführt werden. Mit Benennung sind alle Mechanismen gemeint, mit denen Daten identifiziert und adressiert werden. Die Fähigkeiten von Grundschulern auf diesen Gebieten wurden mit Hilfe von Spielen gemessen, die ohne Computer – nur mit Buntstiften und Papier - durchgeführt wurden. Die Ergebnisse dieser Studien verwenden wir an der Universität Münster als Grundlage für das Design multimedialer Online-Spiele zur Algorithmik für Grundschüler. Am Ende dieses Artikels wird ein Beispiel vorgestellt.

2 Algorithmen im Alltag von Grundschulkindern

Unser Leben ist durchdrungen von Algorithmen. Kommandos wie „Sei jetzt still!“ oder komplexere Instruktionen mit Kontrollstrukturen wie „Wenn du bis Hundert gezählt hast, kannst du mit dem Zähneputzen aufhören“ kennen wir von klein auf. Nach dem Kommunikationsmodell von Schulz von Thun haben sprachliche Äußerungen (Nachrichten) implizit immer auch appellativen Charakter [ST01]. Wenn man in einer zwischenmenschlichen Kommunikationssituation etwas sagt, möchte man in der Regel sein Gegenüber auch dazu bewegen, etwas zu tun. In welchem Ausmaß gehören Texte mit *explizit* algorithmischem Inhalt zum Alltag von Kindern im Grundschulalter? Im Dezember 2008 habe ich 126 Kinder im Alter von 8 bis 10 aus dritten und vierten Klassen dreier Grundschulen in Münster und Witten zu diesem Thema befragt. Darunter waren 67 Mädchen und 58 Jungen (ein Kind hat keine Angaben zum Geschlecht gemacht). Tabelle 1 zeigt einige Ergebnisse. Die meisten Kinder dieser Altersgruppe haben sich bereits mit algorithmischen Dokumenten wie Spiel- oder Bastelanleitung auseinandergesetzt.

Aktivität	Häufigkeit			
	Noch nie	Ein oder zwei Mal	Öfter als zwei Mal	Keine Angabe
Ein Modell aus einem Überraschungsei nach Anleitung selbst gebaut	10	26	88	2
Etwas aus Lego nach einer Anleitung gebaut	17	31	77	1
Spielregeln eines Spiels gelesen	11	38	77	0
Selbst jemandem ein Spiel erklärt	0	24	102	0
Nach einer Bastelanleitung etwas gebastelt	13	46	66	1

Nach Rezept etwas gekocht	28	39	58	1
---------------------------	----	----	----	---

Tabelle 1: Rezeption von algorithmischen Texten bei Grundschulkindern. Befragt wurden 126 Schüler/innen aus dritten und vierten Grundschulklassen, Durchschnittsalter 8,9 Jahre, 67 Mädchen, 58 Jungen (ein Kind machte keine Angabe zum Geschlecht).

3 Spiele mit Bleistift und Papier zur naiven Algorithmik

Wenn man aufwändige multimediale Medien entwickelt, muss man wissen, welchen Schwierigkeitsgrad man seiner Zielgruppe zumuten kann. Welche algorithmischen Sprachkonstrukte verstehen Kinder im Alter von acht bis zehn und welche noch nicht? Was sind typische Barrieren? Um mit möglichst wenig Aufwand Antworten zu finden, habe ich zwei Übungen entwickelt, die ohne Computer im Klassenraum durchgeführt werden können.

3.1 Methodik

Die Übungen wurden als Spiele gestaltet, die den Kindern Spaß machen und als bereicherndes Lernerlebnis empfunden werden sollten. Jeder sollte die Aufgaben zumindest grundsätzlich lösen können und zu Erfolgserlebnissen kommen. In beiden Übungen war das Ziel, ein Bild zu erstellen. Dabei folgten die Kinder jeweils einer Anleitung, die unterschiedliche Arten von Instruktionen enthielt. Entscheidungen, was an welche Stelle geschrieben oder gezeichnet wurde oder welche Farben verwendet werden sollten, wurden algorithmisch getroffen. Dabei spielte auch der Zufall eine Rolle, so dass am Ende jedes Bild anders aussah.

Im Briefing vor Beginn des Spiels wurden die Verhaltensregeln geklärt. Gespräche mit Nachbarn waren erlaubt. Allerdings sollte sich jeder nur für seine eigene Zeichnung verantwortlich fühlen und sich nicht aktiv in die Arbeit seiner Mitschüler einmischen – z.B. wenn sie oder er glaubte einen Fehler entdeckt zu haben. Die Kinder wurden darauf hin gewiesen, dass am Ende bei jedem das Bild anders aussieht. Wenn jemand für einen Nachbarn, der nicht weiter kam, eine Aufgabe erledigen wollte, wurde interveniert. In der Tat war die Stimmung sehr entspannt und aktiv. Die Ergebnisse wurden immer wieder verglichen und diskutiert. Da die Kinder kooperierten, repräsentieren die Ergebnisse nicht nur die Performanz eines Individuums, sondern zu einem gewissen Anteil auch die Leistungsfähigkeit der Nachbarschaft im Klassenraum. Die Kinder konnten die gesamte Unterrichtsstunde nutzen. Es gab keinen Zeitdruck. Jeder sollte in Ruhe fertig werden.

Während der Übung waren zwei oder drei Erwachsene im Raum, die Fragen beantworteten – möglichst ohne die Lösung der Aufgabe zu verraten. Wie gesagt – es sollten keine Misserfolgserlebnisse aufkommen. In vier Workshops wurde mitprotokolliert, zu welchen Schritten die Kinder Fragen gestellt hatten. Die Anzahl der Fragen zu einer Aufgabe kann man als Indikator für den Schwierigkeitsgrad betrachten.

3.2 Der Malroboter – Steuerung

Die jungen Teilnehmer/innen der Übung sollten sich vorstellen, ein Roboter zu sein, der ein Bild zeichnet und alles genauso macht, wie es sein Programm vorschreibt.

Das „Programm“ stand auf einem Zettel und war eine Folge von umgangssprachlich formulierten Anweisungen. Sie enthielten die wesentlichen Kontrollstrukturen einer Programmiersprache: Einseitige und zweiseitige Programmverzweigungen (if-Anweisung, if-else-Anweisung) sowie Iterationen, d.h. durch Daten gesteuerte Wiederholungen wie z.B. „Zeichne an jede Spitze des Sterns einen Kreis“. Diese Kontrollstrukturen wurden mit Sprachkonstrukten formuliert, die man aus dem Alltag kennt. Etwas ungewohnt für die Schüler/innen war die Beschreibung einer Wiederholung, die durch eine Bedingung gesteuert wird:

Schritt 10: Male eine Blume. Schritt 11: Wenn auf deinem Blatt weniger als drei Blumen sind, gehe zu Schritt 10. Sonst gehe zu Schritt 12.

Die letzten Instruktionen des Programms verlangten eine Bewertung der Übung. Dabei sollten die Schüler aus einer Folge von Textelementen eines auswählen und an eine bestimmte Stelle auf ihr Bild schreiben. Beispiel:

Schritt 12: Wie schwierig findest du es, Malroboter zu spielen? Schreibe als Antwort unten auf das Blatt eines der folgenden Wörter: superleicht, leicht, mittel, schwierig, superschwierig.

Tabelle 2 zeigt einen Teil der Auswertung. Es zeigte sich, dass die überwiegende Mehrheit aller Kinder die Übung problemlos bewältigen konnte. Eine etwas höhere Fehlerrate gab es bei Instruktionen mit komplexeren Bedingungen und bei den Bewertungsaufgaben am Ende.

Anweisung	Korrekte Lösungen	Prozent
Zeichne einen roten oder blauen Stern	117	93.6%
Zeichne an jede Spitze des Sterns einen kleinen Kreis.	123	98.4%
Wenn du ein Mädchen bist, male rechts oben auf das Blatt einen Dreieck.	121	96.8%
Schreibe eine Zahl zwischen 1 und 10 neben den Stern	116	92.8%
Wenn deine Zahl kleiner als 5 ist, male einen Kreis um die Zahl, sonst male ein Viereck um die Zahl.	112	89.6%
Wenn dein Vorname ein A oder E enthält, unterstreiche deinen Vornamen.	113	90.4%

Wenn dein Stern auf dem Blatt rot ist und dein Vorname unterstrichen ist, male neben deinen Vornamen ein Herz.	106	84.8%
Schritt 10: Male eine Blume. Schritt 11: Wenn auf deinem Blatt weniger als drei Blumen sind, gehe zu Schritt 10, sonst zu 12	112	89.6%
Schritt 12: Wie schwierig findest du es, Malroboter zu spielen? ... (Korrekte Platzierung des Textes auf dem Blatt)	88	68.8%

Tabelle 2. Korrekte Ausführung von Instruktionen. Auszug aus der Auswertung von 125 Bildern von Grundschulern aus dritten und vierten Klassen.

Der häufigste Fehlertyp war das Ignorieren von Details einer Instruktion. Bei dem Programmschritt, bei dem eine Einschätzung der Schwierigkeit auf das Blatt geschrieben werden sollte, hatten 23.4 % den Text nicht an die richtige Stelle auf das Blatt geschrieben. 3.1% hatten den Text abgewandelt (z.B. „Ich fand es superleicht!“). Dieses Detail wurde entweder übersehen oder als unwichtig erachtet und deshalb nicht ausgeführt. Die meisten Schülerinnen bezeichneten die Übung als leicht (62 wählten das Attribut „superleicht“, 42 „leicht“, 12 „mittel“ und niemand wählte „schwierig“ oder „superschwierig“). Fast allen machte die Übung Spaß. Nur ein einziger Schüler (von 125) fand sie langweilig.

3.3 Benennung von Entitäten

Daten, die in einem Algorithmus verarbeitet werden, müssen irgendwie benannt werden. Namen spielen im menschlichen Denken und Sprechen eine entscheidende Rolle. Grundsätzlich haben Namen zwei Funktionen: Identifizierung (Unterscheidung der benannten Entität von anderen Entitäten) und Adressierung (Ermöglichung des Zugriffs). Menschliche Sprachen (verbal und nonverbal) enthalten eine Reihe von Benennungskonzepten. Eine Entität kann - nonverbal - durch eine Geste identifiziert und adressiert werden (auf etwas zeigen, einer Person in die Augen sehen, die rechte Hand zum Gruß ausstrecken). Verbal geäußerte Namen können direkt und explizit sein, wie die Anrede einer Person mit ihrem Namen. Namen können auch indirekt sein und aus anderen Namen mit Hilfe zusätzlicher Information konstruiert werden (die zweite Person von links auf einem Foto, die jüngste Tochter meines Bruders).

Die Fähigkeit Namen in Dokumenten zu verarbeiten kann als Teil allgemeiner Lesekompetenz betrachtet werden. 20% der Aufgaben aus der internationalen PIRLS-Studie, die die Lesekompetenz von Schülerinnen und Schülern aus vierten Schuljahren untersucht, verlangen direkte Informationsgewinnung aus Sachtexten. Das folgende Beispiel bezieht sich auf den Text „Introducing Antarctica“ [M07], der aus mehreren relativ unabhängigen Abschnitten mit eigenen Überschriften besteht.

„Which section of the article tells you how thick the ice is in Antarctica?“

Aus informatischer Sicht ist hier nach dem Namen einer Kollektion von Wissens-elementen gefragt, die ein gegebenes Wissens-element (Eisdicke) als Item enthält. Der gesuchte Name ist die Überschrift eines Abschnitts. Sie identifiziert den Abschnitt. Da das Wort „thick“ auch im Text vorkommt, braucht eine Schülerin nur nach dieser Zeichenkette zu suchen und dann die Überschrift nachlesen.

3.4 Der Zufallspark

In der Übung „Zufallspark“ zeichnen die Schüler einen Park mit Wegen, Schildern, Tieren, Schaukeln, Mülleimern und anderen Dingen. Sie erhalten eine Anleitung (Programm) und einen Zettel mit Tabellen und Feldern, in die die Spieler Daten eintragen können (Abb. 1).

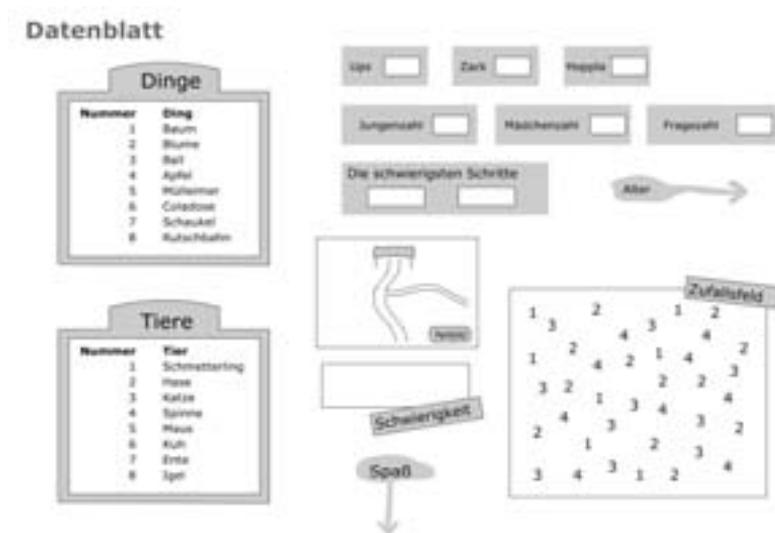


Abbildung 1. Das Datenblatt der Übung „Zufallspark“.

Namen tauchen sowohl in den Anweisungen des Programms als auch auf dem Datenblatt auf. Auf dem Datenblatt wird die Beziehung zwischen Name und benannter Entität auf zweierlei Weise dargestellt: (1) Der Name ist ein Etikett das an der Entität angebracht ist (z.B. Zufallsfeld), oder die Beschriftung eines Behälters (Feld, Rahmen), in den Daten eingetragen werden können (z.B. Ups). (2) Zwischen Name und benannter Entität ist ein Pfeil (z.B. Alter).

Tabelle 3 zeigt eine Auswahl von Instruktionen aus der Anleitung des „Zufallsparks“. Die Übung wurde in sieben Workshops in fünf vierten und zwei dritten Grundschulklassen durchgeführt.

Beteiligt waren 149 Kinder, Durchschnittsalter 9.2 Jahre. In der zweiten Spalte steht die Anzahl der Kinder, die die jeweilige Aufgabe völlig richtig gelöst hatten. Die Schüler hatten während der Übung Gelegenheit Fragen zu stellen. Die Hilfestellung beschränkte sich auf strategische Hinweise wie „Lies noch mal den ersten Satz“ oder „Schau dir das Datenblatt noch einmal genau an.“ Bei fünf Workshops mit insgesamt 107 Schülerinnen und Schülern wurde von den Betreuern (zwei oder drei Erwachsene) protokolliert, zu welchen Schritten ihnen Fragen gestellt wurden. In der dritten Spalte der Tabelle steht die Anzahl der Fragen, die zu einer Instruktion insgesamt gestellt wurden.

Anweisung	Richtig (n=149)	Fragen (n=107)
Schritt 3: Auf dem Datenblatt ist eine Tabelle mit Tierarten. Zeichne an den Blumenweg das Tier mit der Nummer 2.	126	9
Schritt 7: Frage eine Mitschülerin (Mädchen) nach einer beliebigen Zahl zwischen 5 und 9. Das ist die Mädchenzahl. Schreibe die Zahl an die richtige Stelle auf das Datenblatt.	146	15
Schritt 10: Auf dem Datenblatt ist das Zufallsfeld. Nimm einen roten Buntstift und mache mit geschlossen Augen auf dem Zufallsfeld einen Punkt. Die Zahl, die dem Punkt am nächsten ist, heißt Ups. Schreibe die Zahl an die richtige Stelle auf das Datenblatt.	140	9
Schritt 12: Rechne Ups plus Zack. Das Rechenergebnis heißt Hoppla. Schreibe diese Zahl an die richtige Stelle auf das Datenblatt.	142	0
Schritt 13: Schreibe dein Alter (z.B. 11 Jahre) an die richtige Stelle auf das Datenblatt.	137	6
Schritt 14: Zeichne das Ding mit der Nummer Fragezahl rechts neben den Hauptweg.	105	12
Schritt 15: Welches Ding hat die gleiche Nummer wie die Maus? Zeichne dieses Ding unter den Blumenweg.	101	6
Schritt 17: Zeichne das Ding mit der Nummer Ups + 1 mitten auf den Hauptweg.	79	7
Schritt 18: Ein Bubu ist ein Ding mit der Nummer Jungenzahl. Zeichne eine Reihe aus Mädchenzahl Bubus.	24	43

Tabelle 3: Einige Instruktionen aus dem Spiel „Zufallspark“, richtige Lösungen von 149 Kindern aus dritten und vierten Klassen und Anzahl der Fragen, die in fünf Workshops mit insgesamt 107 Kindern gestellt worden sind.

In den Instruktionen werden verschiedene Benennungstechniken verwendet. Explizite Namen können willkürlich gewählt sein (Ups, Zack, Hoppla). Sie können aber auch eine Bedeutung haben, die zusätzliche Information über die benannte Entität liefert. Zu welchem Datentyp gehört sie? Welche Rolle spielt sie im algorithmischen Kontext? Auf welche Aspekte der Realwelt bezieht sie sich? Im Schritt 7 (siehe Tabelle 3) gibt der explizite Name „Mädchenzahl“ einen Hinweis, dass die Entität eine Zahl sein soll und deutet gleichzeitig den algorithmischen Kontext an, da diese Zahl von einem Mädchen erfragt worden ist. Im Schritt 10 korrespondiert der Name *Zufallsfeld* mit der Rolle des benannten Bildes im algorithmischen Kontext während der Name *Ups* offensichtlich keine Semantik hat.

Ein indirekter Name wird aus anderen Namen konstruiert. So bezeichnet die Formulierung „das Tier mit der Nummer 2“ (siehe Schritt 3) das zweite Element der Tabelle Tiere. Komplexere Beispiele findet man in den Schritten 12 und 15 des Zufallszoos.

3.5 Diskussion

Wie gut gelingt es den Grundschulern Namen in verschiedenen algorithmischen Kontexten zu verstehen und zu verarbeiten?

Benennung. Wenn eine irgendwie beschriebene Entität (z.B. ein Rechenergebnis) benannt wird, dominiert die Funktion der Identifikation. Sofern man sich in einem solchen Kontext um einen sinnvollen Namen bemüht, versucht man das Besondere der benannten Entität, das was sie von anderen unterscheidet, zu finden und im Wortlaut des Namens zum Ausdruck zu bringen. Ein Beispiel für eine Benennung ist Schritt 7: „Frage eine Mitschülerin (Mädchen) nach einer beliebigen Zahl zwischen 5 und 9. Das ist die Mädchenzahl. ...“. Wenn ein Schüler auf dem Datenblatt in das Feld mit dem Namen Mädchenzahl (siehe Abbildung 1) eine Zahl zwischen 5 und 9 einträgt, hat er die Benennung verstanden. Offenbar kamen die beobachteten Grundschüler sehr gut mit allen vorkommenden Benennungskonzepten zurecht. Dabei scheint es relativ unerheblich zu sein, ob explizite Namen sinnvoll (z.B. Schritt 7) oder sinnlos (z.B. Schritt 12) sind, und ob für die Visualisierung der Zuordnung eines Namens zu einer Entität Pfeile (z.B. Schritt 13) oder Etiketten an Behältern (z.B. Schritte 7, 10, 12) verwendet werden.

Einfacher Zugriff. Im „Zufallspark“ wird an einigen Stellen eine Entität beschrieben, die auf dem Datenblatt gefunden werden muss, z.B. in Schritt 3 „...das Tier mit der Nummer 2 ...“. Diese Phrase ist hier ein (indirekter) Name für die Zeichenkette „Hase“. Adressierungsaufgaben, bei denen nur ein expliziter Name vorkam und ein einziger Zugriff auf eine benannte Entität vorkam (z.B. Schritt 3), bereitete den meisten Workshopteilnehmern keine Schwierigkeiten. Die Nutzung von Namen zum Zugriff auf Daten scheint Kindern dieser Altersgruppe grundsätzlich vertraut zu sein.

Namen als Platzhalter. Ernsthaftige Probleme traten erst auf, wenn in einer zusammenhängenden algorithmischen Formulierung mehrere Namen vorkamen und benannte Entitäten verarbeitet werden mussten (Schritte 17 und 18). Obwohl die Einzeloperationen – Benennung und Adressierung – den Schülern vertraut sind, scheitern viele bei ihrer Kombination in einem zusammenhängenden Algorithmus. Warum? Möglicherweise fehlt ihnen noch das Platzhalterkonzept, das in diesem Zusammenhang hilfreich sein kann. Die sprachliche Formulierung „Zeichne das Ding mit der Nummer Ups + 1“ (Schritt 17) ist für sich genommen zunächst sinnlos. Eine Möglichkeit, sie zu interpretieren ist, die vorkommenden Namen als Platzhalter zu betrachten und schrittweise durch die benannten Entitäten zu ersetzen.. Aus „Ding mit der Nummer Ups + 1“ wird „Ding mit der Nummer 2 + 1“, dann „Ding mit der Nummer 3“ und schließlich „Ball“. Möglicherweise fehlt Grundschulkindern die Kompetenz zu solchen gedanklichen Umformulierungen.

4 Algorithmische Spiele für Grundschul Kinder

4.1 Konstruktivismus versus Museumspädagogik

Die Idee der Informatik für Kinder ist schon fast so alt wie die Informatik selbst. Bereits im Jahre 1967 – zu einer Zeit in der Computer noch ganze Räume füllten – veröffentlichte Seymour Papert die Programmiersprache Logo. Seitdem sind viele Varianten konstruktivistisch orientierter Entwicklungsumgebungen für Kinder entstanden, darunter die Logo-basierten Microworlds (<http://www.microworlds.com/>) und Scratch (<http://scratch.mit.edu/>). Das gemeinsame Ziel ist, Kindern eine Werkstatt zu bieten, in der sie kreativ ihre algorithmischen und gestalterischen Ideen realisieren können. Ganz nebenbei – so die Idee des Konstruktivismus – werden Kompetenzen insbesondere im Bereich des formalen Denkens entwickelt.

Die Applikationen, die wir in Münster entwickeln, unterscheiden sich von diesem Ansatz – ja, sie sind geradezu ein Gegenpol. Wir wollen keine Werkstatt, wir wollen ein Museum. Unsere multimedialen Algorithmik-Spiele sind als Exponate eines „virtuellen Museums der Informatik“ gedacht, an dessen Web 2.0-Plattform gerade gearbeitet wird. Wie in einem richtigen Museum „betrachtet“ ein Besucher ein Exponat nur für einige Minuten und geht dann weiter. Die Applikationen sind zwar interaktiv aber es geht nicht um schöpferisches Konstruieren. Die Absicht ist vielmehr die Aufmerksamkeit der Besucher gezielt auf Themen der Informatik zu lenken und sie in eine intellektuelle Auseinandersetzung zu verstricken [We07]. Um Missverständnissen vorzubeugen: Der pädagogische Wert des Konstruierens soll hier keinesfalls in Frage gestellt werden. Aber Phasen mit anderen Formen der aktiven Elaboration informatischer Inhalte können eine Ergänzung sein.

4.2 Informatik im Kontext

In der Naturwissenschaftsdidaktik kennt man seit der Mitte der 1980er Jahre ein Unterrichtsmodell, das unter dem Namen „Science in Context“ oder „Salters Approach“ bekannt geworden ist [B02, BL06]. Kontexte sind alltagsrelevante und lebensweltbezogene Fragestellungen, die fachwissenschaftliche Inhalte in einen für Schüler relevanten und einsichtigen Sinnzusammenhang stellen. Der Kontext liefert oft den subjektiven Sinn, die Motivation, sich mit einem fachlichen Inhalt zu beschäftigen. Nach dem Modell der Interessensgenese von Krapp kann sich aus einem in einem konkreten Handlungsablauf erlebten situativen Interesse ein dauerhaftes individuelles Interesse entwickeln – verbunden mit der Bereitschaft sich mit dem neuen Gegenstand intensiver auseinanderzusetzen [K98]. Das impliziert, dass interessante Kontexte zum Aufbau von langfristigem Interesse für informatische Inhalte führen können. In der internationalen ROSE-Studie (Relevance of Science Education), die die naturwissenschaftlichen Interessen von Fünfzehnjährigen untersucht hat, wurden Inhalte *und Kontexte* untersucht und als Dimensionen der Itemkonstruktion verwendet [E07, HB07].

Nun sind im Bereich der Informatik genuine *Inhalte* oft relativ abstrakt und weit von der Lebenswirklichkeit und dem Interessenshorizont junger Menschen entfernt. Deshalb kommt in der Informatik der Konstruktion motivierender *Kontexte* besondere Bedeutung zu. Mit Hilfe multimedialer Technik ist es möglich, Handlungszusammenhänge zu schaffen, die gezielt Kinder bestimmter Altersgruppen ansprechen und faszinieren. Sie können an die Realität anknüpfen und lebensnah sein, müssen es aber nicht. In Hongkong hat man gute Erfahrungen mit mathematischen Lernspielen gemacht, die in chinesische Märchenszenarien eingebettet sind [L03]. Gerade *weil* die Inhalte der Informatik abstrakt sind, hat man beim Design interessanter Kontexte nahezu grenzenlosen Gestaltungsfreiraum.

4.3 Wie betreut man einen Außerirdischen?

Als Beispiel stelle ich in diesem Abschnitt ein Fantasie-orientiertes multimediales Spiel für Grundschüler ohne Programmierkenntnisse vor. Es ist als Exponat für das „virtuelle Museum der Informatik“ der Universität Münster gedacht, ist aber bereits jetzt in einer (englischsprachigen) Stand-alone-Version im WWW öffentlich verfügbar (www.c-park.org/vmi).

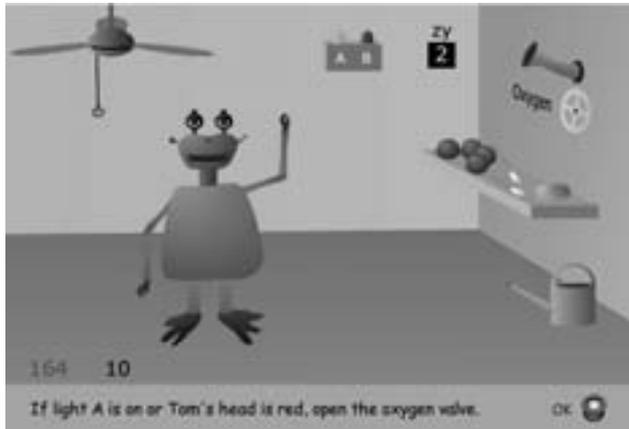


Abbildung 2. Screenshot aus dem Spiel „Take Care of an Alien“ (internationale Version)

Das Spielszenario ist die Lounge eines Weltraumhafens. Die Spielerin bzw. der Spieler hat die Aufgabe, einen Außerirdischen namens Tom zu betreuen, der auf seinen Anschlussflug wartet. Am unteren Bildrand erscheinen nach und nach Anweisungen bzw. kurze Algorithmen, die auszuführen sind. Sie enthalten – wie bei den Bleistift- und Papier-Spielen – verschiedene Kontrollstrukturen und Benennungen. Beispiele: „Wenn Tom einen roten Kopf hat, gieße ihm Wasser über den Kopf bis er wieder grün wird.“ oder „Gib Tom zy Äpfel zu essen.“

Wenn der Knopf „ok“ in der rechten unteren Ecke gedrückt worden ist, wird geprüft, ob das Programmfragment richtig abgearbeitet worden ist (in diesem Fall gibt es Punkte) und es kommt die nächste Anweisung. Das Ziel ist, innerhalb von fünf Minuten möglichst viele Instruktionen korrekt auszuführen.

5 Fazit

Das Erfinden, Formulieren und Anwenden von Algorithmen gehört zu den Kulturtechniken einer modernen Gesellschaft. Die in diesem Beitrag vorgestellten Medien und Unterrichtsaktivitäten zeigen Ansätze, wie man bereits in der Grundschule die Entwicklung algorithmischer Kompetenzen anregen kann. Ab der dritten Klasse scheinen Kinder keine grundsätzlichen Schwierigkeiten zu haben, die wesentlichen Ausdrucksmittel einer Programmiersprache zur Steuerung von Aktivitätsflüssen und zur Benennung von Entitäten zu begreifen und anzuwenden. Für die Gestaltung von weiterführendem Informatikunterricht in den Sekundarstufen ist es wichtig zu wissen, was Schüler/innen an naiver Algorithmik bereits beherrschen und was nicht. Auf welche Konzepte kann man aufbauen? Welche Konzepte müssen erst behutsam eingeführt werden, um Überforderungen zu vermeiden? Welche zum Teil unbewussten intuitiven Vorstellungen können zu Blockaden für das weitere Lernen führen? Um besseren Informatikunterricht zu machen, brauchen wir noch viel detailliertes Wissen über die Denkweise von Kindern im Zusammenhang mit algorithmischen Problemen.

Und nochmals: Algorithmik ist für alle wichtig. Denn letztlich geht es um eine Kompetenz, die nicht nur den Umgang mit Maschinen betrifft sondern auch für das menschliche Zusammenleben grundlegend ist: die Fähigkeit anderen mitzuteilen, was man will und zu verstehen, was die anderen von einem wollen.

Literaturverzeichnis

- [B02] Bennet, J.; Holman, J.; Lubben, F.; Nicolson, P.; Prior, C.: Science in Context: The Salters Approach. Contribution to the 2nd IPN-YSEG-Symposium; 2002.
- [BL06] Bennet, J.; Lubben, F.: Context-based Chemistry: The Salters Approach. In: International Journal of Science Education, 28/9, 2006; S. 999–1015.
- [E07] Elster, D.: In welchen Kontexten sind naturwissenschaftliche Inhalte für Jugendliche interessant? Ergebnisse der ROSE-Erhebung in Österreich und Deutschland. In: Plus Lucis – Zeitschrift der physikalisch-chemischen Gesellschaft in Österreich (3), 2007; S. 2-8.
- [GI08] Arbeitskreis "Bildungsstandards" der GI (Hrsg.): Grundsätze und Standards Informatik für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. Beilage zu LOG IN, 28. Jg. (2008), Heft Nr. 150/151.
- [HB07] Holstermann, N; Bögeholz, S.: Interesse von Jungen und Mädchen an naturwissenschaftlichen Themen am Ende der Sekundarstufe I. In: Zeitschrift für Didaktik der Naturwissenschaften; Jg. 13, 2007; S. 71–86.
- [K98] Krapp, A.: Entwicklung und Förderung von Interessen im Unterricht. In: Psychologie in Erziehung und Unterricht, 44. Jahrgang, Ernst Reinhardt Verlag München Basel, 1998; S. 185-201.
- [L03] Lee, F.-L.; Lee, J.; Lau, T.: Fantasy-based Learning on the Web - Tong Pak Fu and Chou Heung: the Probabilistic Fantasy, Hong Kong Science Teachers' Journal, 21, 2003; S. 10-14.
- [M07] Mullis, I. V. S.; Martin, M. O.; Kennedy, A. M.; Foy, P.: PIRLS 2006 International Repport. IEA's Progress in Intenational Reading Literacy Study in Primary Schools in 40 Countries. TIMMS & PIRLS International Study Center, Lynch School of Education., Boston College 2007.
- [ST81] Schulz von Thun, F.: Miteinander reden – Störungen und Klärungen. Rowolt. Reinbeck bei Hamburg; 35. Aufl. 2001.
- [We07] Weigend, M.: Logo Nanoworlds. In: Proceedings EuroLogo 2007 Bratislava; URL: <http://www.di.unito.it/%7Ebarbara/MicRobot/AttiEuroLogo2007/proceedings/P-Weigend.pdf>
- [W06] Wing, J. M.: Computational Thinking. Communications of the ACM Vol. 49, No. 3, 2006; S. 33–35.

Informatische Konzepte mit Robotern vermitteln – Ein Unterrichtsprojekt für die Sekundarstufe I

Markus Weber, Bernhard Wiesner

Didaktik der Informatik
Universität Erlangen-Nürnberg
Martensstr. 3
91058 Erlangen
markus.weber.nbg@gmx.de
wiesner@informatik.uni-erlangen.de

Abstract: Im Vordergrund der bisher veröffentlichten Projekte zur Nutzung von Robotersystemen aus dem Bildungs- und Freizeitbereich in der Sekundarstufe I steht vorwiegend der Roboter als Untersuchungsgegenstand, dessen Eigenschaften und Fähigkeiten von den Lernenden erforscht werden. Die Nutzung von Robotern als Unterrichtsmedien für den Informatikunterricht der Sekundarstufe I wird dagegen nur vereinzelt beschrieben. Am Beispiel eines Unterrichtsprojekts wird gezeigt, dass diese Robotersysteme geeignet sind, das Vermitteln einer Reihe informatischer Konzepte zu unterstützen, welche bisher mit weniger praktisch orientierten Methoden unterrichtet wurden. Das beschriebene Projekt zeichnet aus, dass es bezüglich der Unterrichtsmethodik und dem zu leistenden Organisationsaufwand sehr flexibel gehandhabt werden kann. Thematisch geht es bei dem Projekt um die Nachbildung der Abläufe im Wareneingang eines Betriebs. Mehrere Roboter arbeiten zusammen, um Waren von einem Förderband zu einem Lagerplatz zu transportieren. Dabei steuern sie sich gegenseitig, indem sie Nachrichten austauschen. Die Anordnung lässt sich im Informatikunterricht der Sekundarstufe I in ausgewählten Teilen realisieren oder im Ganzen als größeres Projekt gestalten.

1 Motivation

Im vorliegenden Projekt werden Robotersysteme zusammen mit einer ikonischen Programmierumgebung als Unterrichtsmedien benutzt, um die Vermittlung ausgewählter Inhalte des Informatikunterrichts in der Sekundarstufe I zu unterstützen.

Unterrichtliche Versuche mit Robotersystemen im Informatikunterricht der Sekundarstufe I, ebenso wie außerunterrichtliche Aktivitäten in der betreffenden Altersstufe, sind mehrfach beschrieben. Dabei wird jedoch überwiegend nicht auf die Vermittlung informatischer Konzepte explizit abgezielt, sondern vielmehr auf das Gewinnen einer breiten Palette von Bildungszielen, zusammen mit einer generell besseren Einstellung der Jugendlichen gegenüber der Technik. Dieser Ansatz beruft sich auf den *Konstruktionismus* Seymour Paperts und findet in zahlreichen Curricula zur Robotik seine Umsetzung. Das

Arbeiten mit Robotersystemen soll Kompetenzen in Fachdisziplinen wie Mathematik, Physik und Technik und Informatik und ebenso in überfachlichen Bereichen wie Problemlösen, Kreativität, Teamarbeit schaffen. Entsprechende Evaluationen sind jedoch wenige zu finden ([FM02], [An05]).

Dem hier vorgestellten Projekt liegt ein vergleichsweise komplexes Szenario zugrunde, das als narrativer Anker für den Einstieg in eine Reihe von Themen benutzt wird: Mehrere Robotereinheiten bilden gemeinsam Abläufe der Warenannahme in einem Betrieb nach. Dabei sind die Aufgaben auf die einzelnen Roboter verteilt – der Gesamtablauf wird sichergestellt durch den Austausch von Nachrichten zwischen den Einheiten. Davon ausgehend lässt sich eine Reihe von Lerneinheiten bilden, mit denen mehrere informatische Konzepte erarbeitet werden können. Deren Ausgestaltung kann weitgehend der pädagogischen Situation entsprechend flexibel gestaltet werden. Es ist denkbar, dass Lernende Teile des Gesamt szenarios im Unterricht nachbauen, Teilabläufe modellieren und mit Hilfe der Programmierumgebung realisieren. Auch über Präsentationen, Lehrerexperimente oder das Analysieren fertiger Abläufe lassen sich die Inhalte vermitteln. Unterrichtsprojekte, wie das folgende, in denen die Lernenden arbeitsteilig die Funktionen der einzelnen Einheiten konzipieren, implementieren und am Ende zu einem Gesamtsystem zusammenfügen, erlauben ein hohes Maß an Selbststeuerung, verbunden mit den zahlreichen Vorteilen schulischer Projektarbeit [Hu00]. Gleichzeitig werden die Roboter in diesem Fall als Modelle betrieblicher Komponenten begriffen, deren Relevanz insbesondere für Schülerinnen und Schüler der Realschule unmittelbar einsichtig ist. Ein zusätzlicher Vorteil liegt in der mehrfachen Verwendbarkeit einer bestimmten physikalischen Anordnung. Die Lernenden nutzen die Roboter im Lauf der Zeit immer selbstständiger, anfängliche Nutzungsprobleme verschwinden. Auch bei den Lehrpersonen wächst die Vertrautheit mit dem Medium und gleichzeitig erscheint der organisatorische Aufwand geringer. Allen unterrichtlichen Varianten gemeinsam ist das Warenannahme-Szenario als immer präsenter Ankerpunkt.

2 Unterrichtsbeispiele mit Robotern

Im Folgenden werden einige ausgewählte Veröffentlichungen zusammengefasst, die sich mit informatischen Konzepten befassen, welche mit Robotersystemen als Unterrichtsmedien erfolgreich vermittelbar sind. Bei der Auswahl wurden Projekte berücksichtigt, in denen die Lernenden Programme nicht in Textform kodieren. Die Fähigkeit mit einer textuellen Programmiersprache umzugehen, soll keine Voraussetzung für den Erwerb anderer informatischer Kompetenzen innerhalb der Sekundarstufe I darstellen (vgl. [GI08, S. 35]). Von der Kodierung in Textform können zudem Schwierigkeiten ausgehen, die vom Verständnis der eigentlichen Lernziele weitgehend unabhängig sind. Bei den ausgewählten Projekten erfolgt die Roboterprogrammierung über Symbole (Icons), die aneinandergereiht eine imperative Programmstruktur ergeben.

Mit der Vielzahl möglicher informatischer Inhalte, welche mit Robotersystemen vermittelbar sind, befasste sich eine Arbeit der Didaktikgruppe der Universität Paderborn [MRH00]. Anhand einer Unterrichtseinheit, in der die Lernenden an einem Roboter arbeiteten, der einen Ball in einen Korb werfen konnte, wurde ausführlich auf die infor-

matischen Konzepte eingegangen, die sich mit einem derartigen Vorhaben vermitteln lassen. Dies begann beim Modellierungsprozess, bei dem zunächst die Technik des Roboters, die Struktur des Umfelds und die Möglichkeiten der Software erfasst wurden. Die Teilnehmer entwarfen den Aufbau und definierten notwendige Fähigkeiten eines Roboters, der eine bestimmte Aufgabe zu erfüllen hatte, vorher mit Hilfe von Papiermodellen. Anschließend bauten sie die Roboter auf und arbeiteten an deren Programmierung. Dabei setzten sich die Lernenden mit den Grundelementen der Algorithmen auseinander, sie erlebten die Datenübertragung zwischen PC und Roboter, beziehungsweise die Kooperation zweier Roboter. Auch der zyklische Softwareentwicklungsprozess konnte den Lernenden beim Arbeiten mit den Robotersystemen nahe gebracht werden. Dabei wurden die Roboter so programmiert, dass sie zunächst nur elementare Aufgaben bewältigten, die anschließend schrittweise komplexer wurden. Auch objektorientierte Sichtweisen ließen sich mit den Robotersystemen vermitteln, indem von einem Grundmodell ausgegangen wurde, das für unterschiedliche Aufgabenstellungen ausgebaut und damit wieder verwertet werden konnte.

Im Roboterprojekt einer Berliner Realschule wurde der Themenbereich Algorithmische Grundstrukturen im Informatikunterricht mit Hilfe des Lego-Mindstorms-Systems vermittelt [Te05]. Dabei wurden die Lernenden durch Aufgaben mit ansteigendem Schwierigkeitsgrad an die Thematik herangeführt. Mit Aufgaben, in denen beispielsweise ein Roboterfahrzeug vor einem Hindernis zurückweichen musste, wurde die bedingte Anweisung als algorithmisches Grundelement thematisiert. Der Sinn von Wiederholungen wurde mit Aufgaben erschlossen, in denen die Roboter ein Quadrat umfahren mussten. In Varianten dieser Aufgabenstellungen wurden die Inhalte gefestigt. Gleichzeitig wurde damit nach Ansicht der Lehrperson eine unmittelbare Vorstellung der algorithmischen Grundelemente geschaffen, welche einem späteren Übergang zu Programmiersprachen in Textdarstellung förderlich sein könnte.

An einer Realschule in Erlangen wurde ein Konzept erprobt, in dem ebenfalls von Aufgaben für Robotersysteme ausgehend schrittweise algorithmische Grundstrukturen erarbeitet wurden [WB07]. Den Einstieg in diesen Themenbereich bildete das Formulieren und spielerische Umsetzen von Handlungsvorschriften, bei dem Schülerinnen und Schüler die Rolle von Robotern übernahmen. Das Formulieren von Handlungsvorschriften für Roboter fiel den Lernenden leicht, da sie von vornherein davon ausgingen, dass Roboteraktionen aus fest definierten elementaren Einzelaktionen zusammengesetzt sind. Damit musste der Abstraktionsgrad einer Handlungsvorschrift an dieser Stelle nicht thematisiert werden. Nach dem ersten Kennenlernen der ikonischen Strukturen der graphischen Programmieroberfläche wurde die Darstellung der Grundstrukturen durch Struktogramme erarbeitet. Diese Möglichkeit der formalen Darstellung von Handlungen wurde anschließend benutzt, um Abläufe außerhalb der Roboterumgebung zu modellieren, wie etwa Spielregeln oder Abläufe an Automaten. Schließlich wurde den Lernenden am Beispiel mehrerer Konzepte zur Linienvorfahrung bewusst, dass Algorithmen unterschiedlich schnell zum Ziel führen bzw. nur unter bestimmten Umständen terminieren.

Eine bekannte Maßnahme, um Mädchen den Zugang zur Technik zu erleichtern, stellt das Projekt *Roberta* dar [Mü05]. Diese Kurse werden unter anderem für entsprechende Wahlveranstaltungen an Schulen angeboten. Hierbei werden informatische Konzepte

implizit vermittelt, da die Lernenden bei der Bewältigung von Aufgaben mit den Symbolen der ikonischen Programmiersprache umgehen, welche ihrerseits algorithmische Grundstrukturen repräsentieren. Die Kursleiter entscheiden, inwieweit sie die Inhalte dekontextualisieren und damit die informatischen Inhalte herausarbeiten. In der wissenschaftlichen Begleitforschung zum Projekt *Roberta* wurde die stark motivierende Wirkung des Arbeitens mit den Robotersystemen nachgewiesen [WSW07].

Auch im Informatikunterricht des Gymnasiums wurden entsprechende Unterrichtssequenzen in der Jahrgangsstufe 7 erprobt [St01]. Dabei wurde aus einem einfachen Rollenspiel eine Handlungsbeschreibung in Textform festgehalten. Anschließend stellten die Lernenden über einen intuitiven Zustandsbegriff den Handlungsablauf in einem Zustandsdiagramm formal dar. Dieses Prinzip wurde benutzt, um die Handlungsabläufe bei Roboteraufgaben zu strukturieren und anschließend zu programmieren.

Wie informatische Inhalte in einem *Robotics-Curriculum* eingebettet sein können, macht ein Projekt für 14- bis 16-jährige Schüler einer New Yorker Highschool deutlich [GES04]. In einer Konstruktionsphase verwendeten die Lernenden das LEGO Mindstorms System um Fahrzeuge zu bauen, zunächst ohne Motoren und Controller-Baustein. Damit wurden Versuche auf einer schrägen Ebene durchgeführt und ausgewertet. In der zweiten Phase wurden Motoren und der Controller-Baustein ergänzt. In Form von Aufgaben, die in mehreren Schritten zum angestrebten Ziel führten, lernten die Schülerinnen und Schüler die Programmsymbole einzusetzen. Damit sollte ein implizites Verständnis für die Grundelemente Sequenz, bedingte Anweisung, Wiederholung sowie die Schachtelung dieser Elemente erreicht werden. In einer dritten Phase sollten die erworbenen Kompetenzen genutzt werden, um eine Wettbewerbsaufgabe zu lösen, in der es darum ging, in möglichst kurzer Zeit eine umrahmte Bodenfläche von Hindernissen (in Form von Papierbechern) zu säubern. Solche Wettbewerbsaufgaben sind immer wieder zu finden. Dass sich damit auch informatische Prinzipien gut vermitteln lassen, zeigt die Evaluation eines Unterrichtsprojekts an einer Schule in Brisbane (AU), an dem Schülerinnen und Schüler der 8. bis 10. Klassen teilnahmen. Das Ziel des Kurses war die Bewältigung der beschriebenen Wettbewerbsaufgabe. Die Lernenden waren gehalten, den Algorithmus zunächst schrittweise mithilfe von Ablaufdiagrammen zu modellieren. Es zeigte sich, dass die Mehrzahl der Lernenden den Wert vorheriger Planung erkannte und sich zunutze machte [NMG07, S. 269].

Bei Baukastensystemen lassen sich Roboterfahrzeuge weitgehend variabel gestalten und durch Ergänzen von Sensoren, Aktoren und entsprechender Programmierung mit unterschiedlichen Fähigkeiten ausstatten. Damit lassen sich Prinzipien der Objektorientierung anschaulich machen. Aus der Sicht eines Konstrukteurs kann man sich Modellgenerationen vorstellen, in denen die einzelnen Fahrzeuge von ihren jeweiligen Vorgängern Methoden und Attribute übernehmen (erben) und eigene dazu erhalten [DR01]. Dieses Konzept entfaltet seine didaktische Wirkung allerdings erst dann, wenn zur Programmierung auch eine objektorientierte Sprache verwendet wird, mit welcher der Vererbungsprozess abgebildet werden kann.

Die bisher genannten Beispiele zeigen, dass die Verwendungsmöglichkeiten von Robotersystemen im Informatikunterricht nicht auf die Einführung algorithmischer Grund-

strukturen beschränkt sind. Tabelle 1 gibt eine Zuordnung publizierter Projekte bzw. Projektkomponenten zu den Inhalts- und Prozessbereichen der Bildungsstandards für die Sekundarstufe I [GI08]. Dort wo Projekte mehrere Inhalts- oder Prozessbereiche überdecken, sind jeweils die betreffenden *Projektkomponenten* angegeben.

	Modellieren, Implementieren	Begründen, Bewerten	Strukturieren, Vernetzen
<i>Algorithmen</i>	Beschreiben (Vorhersagen) des Roboterhaltens bei gegebenem Programm [We08]; Aufgaben lösen (Linienverfolger [Te05], Labyrinth [WB07], Fläche säubern [GES04], Hindernissen ausweichen [Mü05, S. 20], etc.); Arbeiten mit Variablen [Mü02], [St01]	Effizienzbewertung von Handlungsvorschriften (z. B. Linienverfolgung [WB07])	Strukturen bilden (Module, Programmblöcke) [We08]; Analysieren von Programmen und Szenen (Dekonstruktion) [MRH00]
<i>Informati- on, Daten</i>	Sensorwerte erfassen und in bedingten Anweisungen benutzen [St01]; Roboter als Objekte [DR01]; Interaktion der Roboter mittels Datenaustausch [MS05], [We08]	Einschränkungen beim Umgang mit Daten [Mü02]	
<i>Informatik- systeme</i>	Entwickeln und Erstellen von Programmmodulen zum Warenan-nahmeprojekt [We08]		Strukturieren des Verhaltens eines Gesamtsystems und seiner Komponenten [We08]
<i>Sprachen, Automaten</i>	Roboterfahrzeuge auf speziellen Fahrflächen [Wi08]; Simulation einer Straßenbahn [We08, S.46]		
<i>Informatik, Mensch, Gesell- schaft</i>		Einfluss der Robo- ter auf die Gesell- schaft [MRH00]	Zusammenwirken von Technik und Informa- tik [We08], [MRH00]

	Kommunizieren, Kooperieren	Darstellen, Interpretieren
<i>Inhalts- übergrei- fend</i>	Entwurf und Konstruktion der Roboter in Partner- arbeit, Erfahrungsaustausch in der Gruppe bzw. Klasse [WB07]; Bearbeiten von Aufgaben mit Wettbewerbscharakter [Te05], [GES04]	Arbeiten dokumentieren; Präsentieren des Szenarios als Abschluss des Unter- richtsprojekts [We08]

Tabelle 1: Zuordnung von Projekten bzw. Projektkomponenten zu Inhalts- und Prozessbereichen der Bildungsstandards [GI08]

Das in den folgenden Abschnitten beschriebene Projekt ist unter der Maßgabe entstanden, möglichst für das Vermitteln mehrerer informatischer Themenbereiche benutzt werden zu können. Der erste Schwerpunkt liegt in der Präsentation der gesamten Anord- nung als Informatiksystem. Die Lernenden verstehen, ausgehend vom Gesamtablauf, die

Funktionen der einzelnen Komponenten und können diese beschreiben. Sie erfahren, dass es sich dabei um ein vernetztes System handelt, dessen Verhalten durch den Austausch von Nachrichten zwischen den einzelnen Komponenten gesteuert wird. Sie lernen, wie ein solcher Nachrichtenaustausch ausgeführt werden kann und implementieren eigene Lösungen mittels Bluetooth-Übertragung. Sie entwickeln geeignete Handlungsvorschriften, implementieren diese mit der ikonischen Programmiersprache und testen sie anschließend. Dabei erfolgt eine intensive Auseinandersetzung mit den algorithmischen Grundelementen sowie ihrer formalen Darstellung. Ein weiterer Schwerpunkt kann auf das Beschreiben des Verhaltens des Gesamtsystems unter Benutzung des Zustandsbegriffs gelegt werden. Die gesamte Anlage, die aus drei oder vier Robotern besteht, kann, falls genügend Zeit zur Verfügung steht, vollständig als Unterrichtsprojekt realisiert werden.

3 Modulares Unterrichtskonzept

Im Folgenden wird das Projekt „Warenannahme“ im Überblick dargestellt, anschließend auf Möglichkeiten der Modularisierung eingegangen. In Abschnitt 4 wird exemplarisch eines der Module detaillierter beschrieben. Das Projekt folgt den Gestaltungsprinzipien des „Anchored-Instruction-Modells“ [SN00]. Im Zentrum des Konzepts steht ein *narrativer Anker*, der als Ausgangspunkt des Unterrichts dient und das Interesse der Lernenden wecken und steuern soll. Dieser Anker besteht aus Situationsbeschreibungen oder Filmen, die in eine Problem- bzw. Aufgabenstellung münden. Ein Teil des Wissens, das Lernende zur Lösung der Problemstellung benötigen, ist in den Ausgangsinformationen enthalten, der Rest muss von ihnen selbst konstruiert werden. Im vorliegenden Projekt stellt ein Videofilm den Einstieg dar. Der Film zeigt einen möglichen Ablauf einer robotergesteuerten Warenannahme (vgl. Abbildung 1). Damit verbunden ist ein Unterrichtsgespräch, in dem die wachsende Bedeutung derartiger Anlagen in Handel und Industrie herausgearbeitet wird. Als Materialien werden den Lernenden Anforderungsbeschreibungen der Problemsituation, Funktionsbeschreibungen der zu verwendenden Robotermodelle, Aufbauanleitungen der Roboter sowie Informationsmaterial zum Nachrichtenaustausch via Bluetooth und Programmieranregungen für die ikonische Programmiersprache zur Verfügung gestellt. Durch diese Methodik sollen sie zu selbstgesteuertem Arbeiten angeregt werden, eine Ansammlung „trägen Wissens“ soll vermieden werden [Re96]. Aus psychologischer Sicht können „Querverbindungen zwischen dem episodischen und dem semantischen Gedächtnis hergestellt“ werden, was „zu einer doppelten Fixierung des Lernstoffs“ führen kann [Hu00, S.10].

Um die eingangs erwähnte Flexibilität zu gewährleisten, wurde das Projekt in einzelne Module gegliedert. Die Module befassen sich jeweils mit grundlegenden Funktionen des Gesamtszenarios. Sie unterscheiden sich darüber hinaus im *Zeitbedarf* (längere Module mit ca. 5 Doppelstunden (DS) und kürzere mit 2 DS), in der *Art und Weise der Schüleraktivität* (Roboterbau, Arbeiten mit Programmen oder Arbeiten mit Video), im *erforderlichen Vorbereitungsaufwand* (Erstellen von Bauplänen, Modifikation der Programme oder Nutzung des Videos), im *Einsatz von Projektmaterialien* (Modul 2 nur Programm der Lagernummerwahl um Variablenkonzept zu verdeutlichen oder Bau ganzer Teile des Projektes im Modul 6) sowie der *fachlich methodischen Intention* (theoretische Untersu-

chung der Zusammenhänge im Modul 2 oder praktische Erarbeitung von Algorithmen zur Linienverfolgung im Modul 1). Nicht zu vermeiden sind dabei Abhängigkeiten oder thematische Überschneidungen zwischen den Modulen. In Tabelle 2 werden zu den Modultiteln die im Vordergrund stehenden informatischen Grundkonzepte sowie eine kurze Beschreibung angegeben.

Modulbezeichnung	Informatisches Grundkonzept	Kurzbeschreibung
Linienverfolgung der Roboter	Grundelemente von Ablaufstrukturen	Zur Linienverfolgung werden von den Robotern die Messwerte zweier Lichtsensoren verglichen und entsprechende Steuerbefehle generiert.
Benutzereingabe der Lagernummer	Variablenkonzept; Datentypen; Variablen als Kontrollstrukturen	Der Benutzer kann eine Lagernummer nur einmal auswählen, das Programm erkennt und speichert bereits gewählte Lagerpositionen in mehreren voneinander abhängigen Variablen.
Auf Signale reagieren	Prozeduren und Funktionen; Grundelemente von Ablaufstrukturen	Straßenbahnmarkierungen dienen neben Bluetooth-Signalen als Auslöser für Roboterverhalten und steuern Funktionen und den Gesamttablauf (z. B. Verlassen einer Warteschleife).
Netzwerke kommunizieren der Roboterheiten	Netzwerkadressierung, Client – Server Prinzip, Informationsübertragung mit Bluetooth, Parallelprozessdatenverarbeitung	Roboterheiten haben spezifische Adressen und senden sich Nachrichten in festgelegte „Mailboxen“, es können damit auch Nachrichten parallel übertragen werden. Eingehende Bluetooth-Nachrichten steuern das Verhalten der Roboter und kontrollieren den Ablauf ihrer Programme (z. B. Starten des Linienverfolgungsalgorithmus).
Codierung und Übermittlung von Information	Codierung von (Bluetooth-) Nachrichten	Informationsübermittlung in den Bluetooth-Nachrichten kann im Klartext (mit kurzen Strings) oder als Zahlencode stattfinden; Recodierung erfolgt programmgesteuert. Vergleich mit Chiffrierungsverfahren (z. B. Caesar-Verschiebung), evtl. Ausblick auf kryptographische Methoden.
Warenübergabe zwischen Roboterheiten	Strukturierung eines Handlungsablaufs; Entwicklung von Problemlösungen	Warenübergabe als fehleranfälliger Vorgang beim Pakettransport erzwingt eine strukturierte und genaue Ablaufplanung; Diskussion der Problemlösungsansätze (Vor-, Nachteile und Auswirkungen).
Warenannahme als informatisches System	Beschreibung und Strukturierung von Handlungsabläufen	Systematische Analyse des Gesamtsystems, Vor-, Nachteile bzw. Grenzen bestimmen; Abhängigkeit von Soft- und Hardware erkennen.

Tabelle 2: Module der robotergesteuerten Warenannahme

4 Beispielmodul „Warenübergabe zwischen Robotereinheiten“

Im Folgenden soll exemplarisch das sechste Modul in seiner Umsetzung detaillierter beschrieben werden. Das Modul „Warenübergabe zwischen Robotereinheiten“ hat seinen Schwerpunkt in der Entwicklung von Handlungsvorschriften für die Übergabe zwischen einem Förderband, auf dem die eingehenden Waren liegen, und einem Transportfahrzeug, das für den Transport zum Lager zuständig ist (vgl. Abbildung 1). Da sich dieser Prozess auf unterschiedliche Art und Weise mit den Robotersystemen realisieren lässt, ergibt sich damit die Möglichkeit der Entwicklung verschiedener Handlungsvorschriften. Diese Varianten können anschließend diskutiert und auf ihre Tauglichkeit getestet werden. In dieser exemplarischen Umsetzung werden das Lego Mindstorms NXT Baukastensystem und die zugehörige ikonische Programmierumgebung verwendet.



Abbildung 1: Modell der robotergesteuerten Warenannahme

Am Ende des Moduls sollen die Lernenden

- mit den algorithmischen Grundelementen sicher umgehen
- Abläufe strukturieren können
- formale Darstellungen von Handlungsvorschriften interpretieren können
- den Nachrichtenaustausch zwischen Robotern implementieren können
- die Rolle des Nachrichtenaustauschs bei vernetzten Systemen verstehen.

Darüber hinaus werden übergeordnete Ziele verfolgt, wie die Fähigkeit zur Präsentation von Arbeitsergebnissen, das Verständnis für die Bedeutung der Arbeitsteilung bei der Bewältigung komplexer Aufgaben sowie Einsicht in die Abhängigkeit von Hard- und Software beim Bau der Roboterkomponenten. Insgesamt sind für dieses Modul 5 DS mit je 90 min vorgesehen. Die geplante Verteilung der Inhalte der Sequenz zeigt Tabelle 3.

Es wird vorausgesetzt, dass den Lernenden die algorithmischen Grundelemente bekannt sind und sie einfache Abläufe formal darstellen können. Sie wiederholen also in dem Modul die Methode des strukturierten Problemlösens. Dadurch wird ein gelerntes Ver-

fahren auf eine neue Situation praktisch angewandt, mit neuen Assoziationen verknüpft und damit vertieft. Das Modul könnte im Anschluss an eine Einführung der algorithmischen Grundstrukturen (z. B. mit Robot Karol) in der 7. Jahrgangsstufe eines Gymnasiums stattfinden. Alternativ ist das Modul auch für die 8. Jahrgangsstufe der Realschule geeignet. Für einen ersten Einstieg in den Bau und die Programmierung der Robotersysteme erscheint das Modul zu komplex, da höhere Anforderungen an das Bauen und Programmieren der Komponenten gestellt werden als etwa bei einem Linienverfolger (Modul 1).

DS	Inhalte
1	Videofilm, Formulierung der Problemstellung durch die Lehrperson Erstellen einer strukturierten Beschreibung des Handlungsablaufs im Video
2	Aufbau des „Transporters“ nach Aufbauanleitung
3	Bau des „Förderbandes“ nach Aufbauanleitung Steuern von Abläufen mit Hilfe der Bluetooth-Nachrichten
4	Diskussion verschiedener Übergabemöglichkeiten der Waren vom Förderband zum Transporter (technische Umsetzung, Konsequenzen für die Programmierung) Lernende erzeugen für eine zugeteilte Übergabe-Variante eine Handlungsvorschrift und implementieren sie.
5	Test der Warenübergabe mit den beteiligten Komponenten durch die einzelnen Gruppen, anschließend Abschlusspräsentation der Ergebnisse

Tabelle 3. Stoffverteilung im Modul „Warenübergabe zwischen Robotereinheiten“

Zu Beginn der ersten Doppelstunde präsentiert die Lehrkraft das Video. Es zeigt, wie ein Paket auf das Förderband gelegt wird und der Benutzer eine Taste drückt, welche die Nummer im Lager angibt. Daraufhin setzt sich das Förderband in Bewegung und sendet an den bereit stehenden Transporter den Auftrag, die Ware aufzunehmen und an eine bestimmte Lagernummer zu transportieren. Sobald das Paket auf dem Transporter liegt, sendet dieser die Nachricht „Paket erhalten“ an das Förderband, das sein Band daraufhin stoppt. Der Transporter bringt danach das Paket an die gewählte Lagerposition.

Danach folgt eine lehrerzentrierte Phase, in der die Hintergründe und Funktionsweisen der robotergesteuerten Warenannahme im Film beschrieben werden. Dazu können mit den Schülerinnen und Schülern die Notwendigkeit und die Vorteile solcher Systeme diskutiert werden. Die Lehrperson kann beispielsweise einen Vergleich zwischen der traditionellen Buchbestellung bei einem Buchladen und einer Buchbestellung im Internet anregen. Dabei sollten die Lernenden erkennen, dass mit der Größe des Warenbestandes ein automatisiertes Lagerverwaltungssystem zeitliche und damit wirtschaftliche Vorteile gegenüber einem traditionell verwalteten Lager hat. Im Anschluss werden die Aufgaben- und Arbeitsteilung bei Lagersystemen erörtert. Die daraus resultierende Warenübergabe zwischen Robotern wird als zentraler Teil der Problemsituation betont. Dies kann man weiter verdeutlichen, indem Berufsbeschreibungen von Lageristen, Lagerverwaltern und LKW-Fahrern eines realen Unternehmens von den Schülerinnen und Schülern verglichen werden. Dieser Vergleich stellt einen Alltags- und Berufsweltbezug für die Lernenden dar und betont die Problemrelevanz. Die Lernenden erhalten die Aufgabe eine mög-

lichst genaue und strukturierte Beschreibung des Handlungsablaufs im Video zu erstellen. Um eine weitgehend formalisierte Darstellung zu erhalten, gibt die Lehrperson Schlüsselbegriffe vor, wie etwa „Förderband“, „Transporter“, „Lagernummer eingegeben“, „Nachricht: Band stoppen“. Jede Gruppe bearbeitet nur einen kleinen Zeitabschnitt des Videos. Hieraus wird am Ende die Gesamtbeschreibung des Ablaufs erstellt.

Für die zweite Doppelstunde erstellt die Lehrkraft eine Aufbauanleitung des Roboters „Transporter“ und stellt als Software das Linienverfolgungsmodul (als „Blackbox“) zur Verfügung. Die Lernenden bauen in Gruppenarbeit diesen Roboter nach, übertragen das Programm und beobachten seine Funktionen in einem Rundkurs. Um Anfangsschwierigkeiten zu minimieren teilt die Lehrkraft einen Leitfaden für die Programmierumgebung aus und führt deren Bedienung mit einer Beamer-Präsentation vor. Mit Hilfe eines Arbeitsblattes, das Aufgabenstellungen und Kurzanleitungen enthält, erstellen die Schülerinnen und Schüler in Partnerarbeit einige Programmkomponenten (z. B. Soundwiedergabe, Textausgabe, Motorenansteuerung u. ä.) und testen diese. Anschließend erhält die Klasse eine dokumentierte Fassung des fertigen Transporter-Programms. Dieses Programm wird in Gruppenarbeit analysiert, verbalisiert und sein Ablauf beschrieben. Das Video und die Beobachtungen beim Rundkurs können dabei das Verständnis erleichtern.

In der dritten Stunde bauen die Schülerinnen und Schüler in Gruppenarbeit das Förderband nach Anleitung auf. Bei der Programmierung stellt sich die Frage, wie lange das Förderband, welches das Paket auf den Transporter lädt, zu laufen hat. Zusammen mit der Lehrkraft werden unterschiedliche Möglichkeiten zur Steuerung der Antriebsmotoren des Förderbandes entwickelt. Lösungen wären beispielsweise, dass die Motoren eine festgelegte Zeit laufen, eine feste Anzahl von Umdrehungen vollführen, die aus dem Umfang des Antriebsrades und der Förderbandlänge ermittelt werden, bis hin zu einem Stopp der Motoren über einen mechanischen Hebel als Auslöser. Nach einer Diskussion der Lösungsansätze lenkt die Lehrperson die Aufmerksamkeit der Klasse wieder auf das Video und zeigt ihr, falls diese Lösung nicht bereits erkannt wurde, die Möglichkeit das Förderband durch eine Bluetooth-Nachricht des Transporters zu stoppen. Die Lernenden sollen im Folgenden mit ihren bisherigen Erfahrungen mit der Programmierumgebung und einem Leitfaden für die Vernetzung der Roboter mittels Bluetooth in Gruppenarbeit eine Lösung für die Nachrichtenübermittlung finden. Zum anschließenden Üben versenden sie mit kleinen Hilfsprogrammen Nachrichten zwischen zwei NXT-Einheiten. Das Senden und Empfangen der Nachrichten sollen dabei die NXT-Bausteine mit Textausgaben am LCD anzeigen (Abbildung 2). Abschließend entwickeln die Gruppen ein Programm zum Stoppen des Förderbandes mittels Bluetooth-Nachricht.



Abbildung 2: LCD-Ausgaben beim Versand/Empfang von Bluetooth Nachrichten

Die vierte Doppelstunde beginnt mit einer Diskussion über verschiedene Möglichkeiten der Paketübergabe (neben der im Film gezeigten Lösung sollen eigene Vorschläge entwickelt werden – z. B. über eine Rutsche). Ziel dieses Teils der Diskussion ist ein Einblick in die Folgen für die Programmierung bei geändertem Aufbau der Paketübergabe. Dazu erörtert die Klasse an ausgewählten Musterbeispielen (z. B. Übergabe mit Greifer oder Schieber) wie die Motorsteuerung (z. B. Greifer: Motorbewegung bis Ware gepackt ist; Schieber: Motorbewegung mit bestimmter Gradzahl) zu realisieren wäre. Außerdem sollte die Lehrperson darauf eingehen, dass es in der Realität Randbedingungen gibt, welche die Möglichkeiten einer maschinengesteuerten Paketübergabe beschränken und dementsprechende Folgen für die Programmierung haben. Beispiele solcher Randbedingungen könnten der Speicherplatzbedarf bestimmter Programmlösungen, die Abmessungen der Waren oder die Motorenleistungsfähigkeit sein. In der Diskussion werden daraufhin die Vor- und Nachteile mehrerer Paketübergaben erörtert und ihr möglicher Einsatz in unterschiedlichen Szenarien erarbeitet. Die Lernenden sollten hierbei beispielsweise erkennen, dass die Randbedingung „Transport zerbrechlicher Waren“ eine Warenübergabe mittels einer steilen Rutsche ausschließt. In der zweiten Doppelstundenhälfte wird jeder Gruppe eine fertig aufgebaute Paketübergabekomponente zugeteilt. Die Gruppen erhalten den Arbeitsauftrag eine Handlungsvorschrift für die Steuerung ihrer Komponente zu entwickeln. Falls zeitlich möglich, sollen sie anschließend damit beginnen, die Vorschrift in ein Programm zu übertragen.

Die letzte Doppelstunde wird von den Gruppen genutzt, um deren Warenübergabekomponente mit den Robotern (Förderband und Transporter) zu verbinden und sie zu testen. Die Lernenden vervollständigen hierzu aus ihrer Handlungsvorschrift das Programm, das die Übergabe steuert und bauen die Übergabekomponente an das Förderband. Im Anschluss an die Testphase erstellt die Gruppe eine kurze Abschlusspräsentation (ca. fünf Präsentationsfolien). Die Präsentation soll eine Beschreibung der Übergabe, Hintergrundinformationen zum mechanischen Ablauf und die Handlungsvorschrift mit deren Implementierung beinhalten, sowie die Gesamtarbeit der Gruppe anschaulich repräsentieren.

5 Fazit und Ausblick

Das beschriebene Projekt beruht auf einer Examensarbeit [We08]. Die darin beschriebene Unterrichtssequenz gibt einen grundlegenden Einblick in das Gesamtsystem und zeigt, wie die angegebenen Unterrichtsziele methodisch erreicht werden können. Vor allem wird deutlich, dass sich mit Hilfe der Robotersysteme wesentlich mehr informatische Konzepte innerhalb des Informatikunterrichts vermitteln lassen, als dies in rein technisch oder spielerisch ausgerichteten Robotik-Kursen bisher geschieht. Voraussetzung ist, dass die Robotersysteme in erster Linie als gezielt einzusetzende Unterrichtsmedien verstanden werden und weniger als Unterrichtsgegenstände, deren technische Details zu ergründen sind. Wünschenswert sind die Fortführung des Projekts in der Entwicklung weiterer konkreter Module sowie anschließende Fallstudien, aus denen weitere Erkenntnisse zur Wirksamkeit des Verfahrens gewonnen werden können.

Literaturverzeichnis

- [An05] Andersen, F. Ø.: "Kids on campus" Optimal learning environments in Japan. Lego Learning Institute, 2005, http://www.legolearning.net/download/Kids_on_campus.pdf (zuletzt geprüft am 15.02.2009).
- [DR01] Dietzel, R.; Rinkens, T.: Eine Einführung in die Objektorientierung mit Lego Mindstorms Robotern. In: Keil-Slawik, R.; Magenheimer, J. (Hrsg.): Informatikunterricht und Medienbildung, INFOS 2001. Köllen, Bonn, 2001; S. 193–199.
- [FM02] Fagin, B. S.; Merkle, L. D.: Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education. In: Journal on Educational Resources in Computing 2 (2002) 4; pp. 1–18.
- [GES04] Goldman, R.; Eguchi, A.; Sklar, E.: Using educational robotics to engage inner-city students with technology. In: ISLS (Hrsg.): Proceedings of the 6th International Conference on Learning Sciences, Santa Monica, California, 2004; pp. 214–221.
- [GI08] Gesellschaft für Informatik (Hrsg.): Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. Beilage zu LOG IN 28 (2008) 150/151.
- [Hu00] Hubwieser, P.: Didaktik der Informatik. 2. Auflage, Springer, Berlin, 2004.
- [MRH00] Magenheimer, J.; Reinsch, T.; Hirsch, M.: Zugänge zur Informatik mit Mindstorms. In: LOG IN 20 (2000) 2; S. 34–46.
- [MS05] Magenheimer, J.; Scheel, O.: Zugänge zur Softwaretechnik. In: LOG IN 25 (2005) 134; S. 39–44.
- [Mü02] Müller, W.: Algorithmik mit dem LEGO-Roboter. Lehrer Online, 2002, URL: <http://www.lehrer-online.de/lego-roboter.php> (zuletzt geprüft am 24.02.2009).
- [Mü05] Müllerburg, M. et al.: Informatikausbildung, Roboter und Mädchen. In: Cremers, A. B.; Manthey, R.; Martini, P.; Steinhage, V. (Hrsg.): INFORMATIK 2005 - Informatik LIVE! Band 1. Köllen, Bonn, 2005; S. 143–147.
- [NMG07] Norton, S. J.; McRobbie, C. J.; Ginns, I. S.: Problem Solving in a Middle School Robotics Design Classroom. In: Research in Science Education 37 (2007) 3; pp. 261–277.
- [Re96] Renkl, A.: Träges Wissen: Wenn Erlerntes nicht genutzt wird. In: Psychologische Rundschau 47 (1996) 2; S. 78–92.
- [SN00] Strittmatter, P.; Niegemann, H. M.: Lehren und Lernen mit Medien. Wissenschaftliche Buchgesellschaft, Darmstadt, 2000.
- [St01] Steuer, H.: Roboter planen, bauen, programmieren. Adam Kraft Gymnasium Schwabach, 2001, URL: <http://www.sc.shuttle.de/sc/akg/robotics/Kurs/index.htm> (zuletzt geprüft am 13.02.09).
- [Te05] Tempelhoff, A.: Robotik in der Sekundarstufe I. In: LOG IN 25 (2005) 134; S. 23–29.
- [WB07] Wiesner, B.; Brinda, T.: Erfahrungen bei der Vermittlung algorithmischer Grundstrukturen im Informatikunterricht der Realschule mit einem Robotersystem. In: Schubert, S. (Hrsg.): Didaktik der Informatik in Theorie und Praxis (GI-Edition Proceedings, 112). Gesellschaft für Informatik, Bonn, 2007; S. 113–124.
- [We08] Weber M.: Vermittlung von informatischen Grundkonzepten der Realschulbildung anhand einer robotergesteuerten Lagerverwaltung. Universität Erlangen-Nürnberg, 2008, URL: <http://ddi.informatik.uni-erlangen.de/Lehre/SA-DA/ddi-robalg.xml> (zuletzt geprüft am 13.02.09)
- [Wi08] Wiesner, B.: Zustandsbasierte Modellierung eines Robotersystems. In: LOG IN 28 (2008) 150/151, S. 29–36.
- [WSW07] Wiesner-Steiner, A.; Schelhowe, H.; Wiesner, H.: The Didactical Potential of Robotics for Education with Digital Media. In: International Journal of Information and Communication Technology Education 3 (2007) 1; pp. 36–44.

Unterschiede im Lernerfolg von Schülerinnen und Schülern in Abhängigkeit von der zeitlichen Reihenfolge der Themen (OOP-First bzw. OOP-Later)

Albrecht Ehlert, Oberstufenzentrum Informations- und Medizintechnik, ehler@oszimt.de
Carsten Schulte, Freie Universität Berlin, schulte@inf.fu-berlin.de

Abstract: Unter den vielen fachdidaktischen Vorschlägen zum Unterrichten der Objektorientierten Programmierung (OOP) im Informatik-Anfängerunterricht werden bei dieser empirischen Untersuchung zwei Wege verglichen, der sog. OOP-First-Einstieg mit dem OOP-Later-Einstieg. Dabei werden zwei Schulklassen ein Schuljahr lang gleichartig in Informatik beschult (nur die zeitliche Reihenfolge der Themen variiert) mit der zentralen Forschungsfrage:
Sind (signifikante) Unterschiede bei den Lernerfolgen festzustellen?
In diesem Beitrag werden einige ausgewählte Ergebnisse dargestellt.

1 Einführung

Obwohl das objektorientierte Paradigma seit über einem Jahrzehnt in den Informatik-Unterricht in Schulen und Hochschulen Einzug gehalten hat, gibt es bis zum heutigen Tag eine offene Diskussion über angemessene didaktisch-methodische Konzepte zur Vermittlung des Paradigmas. Besonders diskutiert wird die Frage, ob die Idee der „objects first“ sinnvoll ist oder nicht. In dieser Diskussion werden viele verschiedene didaktische Dimensionen angesprochen: Angemessene Werkzeuge, Unterrichtsbeispiele, Reihenfolge und Schwerpunkte der Themen, Lehr- und Lernformen (vgl. [Di07], [Ba08], [BK03], [Sp05]). Die Vielfalt und innere Abhängigkeit der diskutierten didaktischen Aspekte verhindert, dass die zentrale Frage „objects first“ oder „objects later“ zu lösen ist. Hier fehlen eine Definition der verfolgten Ansätze sowie eine Abschätzung, welche Aspekte den zentralen Unterschied ausmachen.

In diesem Artikel werden wir zunächst eine solche Definition analytisch entwickeln und durch eine empirische Analyse verschiedener Lehrbücher überprüfen. Anschließend werden aufbauend auf dem so definierten Kernunterschied die beiden Ansätze in einem empirischen Experiment mit zwei Schulklassen in Bezug auf den Lernerfolg verglichen. Dazu werden wir den Aufbau, die Durchführung und die Analyse einer entsprechenden empirischen Studie beschreiben. Anschließend werden einige Ergebnisse in Bezug auf die Konsequenzen für die Schulpraxis und auf weiterführende fachdidaktische Forschungsfragen interpretiert.

2 Von objects-first zu OOP-First

objects first vs. objects later. Barnes und Kölling [BK03] umreißen den „objects-first-Schlachtruf“ folgendermaßen: „Ein Student kann als erste Aktivität ein Objekt erzeugen und seine Methoden aufrufen! Weil Benutzer Objekte direkt erzeugen und manipulieren können, können Konzepte wie Klassen, Objekte, Methoden und Parameter ohne weiteres direkt diskutiert werden, bevor die erste Zeile Java-Quelltext betrachtet werden muss.“ ([BK03], deutsche Ausgabe von 2003, S. 18) Ob diese Idee sinnvoll ist, ist sehr umstritten. So gab es auf der Mailingliste der SIGCSE 2004 einen ausführlichen Streit um Vor- und Nachteile des objects-first-Vorgehens, der hinterher von Fachdidaktikern ausgewertet wurde (vgl. Li06). Es wurde deutlich, dass tatsächlich viele unterschiedliche didaktische Dimensionen angesprochen werden und dass viele der Argumente auf individueller Anschauung und Lehr-Erfahrung beruhen.

In der Zwischenzeit hat es bereits einige empirische Studien gegeben, die den objects-first-Ansatz vergleichend untersuchen. Decker [De03] kommt zum Schluss, dass der objects-first-Ansatz überlegen ist. Allerdings hat sie den Ansatz mit einem sogenannten „Weniger Objekte“-Ansatz verglichen und vor allem bezüglich der so vermittelten objektorientierten Kenntnisse untersucht – da ist es wenig verwunderlich, dass in der Gruppe, in der mehr Zeit für Objektorientierung aufgewendet wurde auch mehr Wissen zum Thema entstanden ist. Andererseits hat sie auch längerfristige Vorteile in nachfolgenden Veranstaltungen festgestellt, und dass die Betonung der Objektorientierung keine Nachteile in den prozeduralen Wissensbeständen ergeben hat. Reges [Re06] kommt dagegen in einem empirischen Vergleich zu dem Schluss, dass die traditionelle Art des Unterrichtens dem objects-first-Ansatz deutlich überlegen sei: Bessere Leistungen, bessere Zufriedenheit der Lernenden, bessere Bewertung des Lehrenden, weniger Abbrecher. Allerdings vergleicht er von ihm selbst durchgeführte Kurse mit denen seines Vorgängers, den die Universität gefeuert hat, um ihn mit der ausdrücklichen Maßgabe anzustellen, die Anfangskurse didaktisch zu verbessern. Da sein Vorgänger im Gegensatz zum ihm einen objects-first-Ansatz vertreten hat, sieht er hier die Hauptursache – es könnte allerdings auch an anderen Dingen liegen: Möglicherweise ist das didaktische Konzept insgesamt besser, der Lehrende fähiger, die Studenten besser (oder besser motiviert), die gestellten Anforderungen geringer, etc.

Diese Studien machen verschiedene Anforderungen an eine empirische Untersuchung des Ansatzes deutlich (und liefern dazu gute Anregungen): A) Wie kann der objects-first-Ansatz angemessen mit einer Alternative verglichen werden? Und B) Welches sind geeignete Messzeitpunkte und Vergleichskriterien?

Ein Aspekt scheint besonders bedeutsam: Was ist der genaue Unterschied der objects-first-Idee im Vergleich zum „traditionellen Unterrichten“?

Unterschiede von objects first und objects later. In einer internationalen Studie, an der auch viele deutsche Informatiklehrerinnen und -lehrer teilgenommen haben, wurden drei Varianten des objects-first-Begriffs unterschieden [BS07]:

1. Objekte benutzen: Wie oben im Zitat angedeutet, werden zunächst vorhandene Objekte benutzt und manipuliert bevor implementiert wird.
2. Klassen schreiben: Von Anfang an werden Klassen definiert, implementiert und instanziiert, um das Paradigma zu vermitteln. Frühe Programmiererfahrungen mit dem Objektorientierten Paradigma stehen im Mittelpunkt.

3. **Konzepte:** In Bezug auf die Diskussion in Deutschland könnte man auch vom objektorientierten Modellieren sprechen. Zunächst werden die abstrakten und generellen Ideen des Paradigmas vermittelt, wobei die objektorientierten Modelle im Mittelpunkt stehen.

Gemeinsam ist den verschiedenen Varianten, dass das objektorientierte Paradigma an den Anfang gerückt wird. Die folgende Abbildung veranschaulicht diesen Zusammenhang am Beispiel der Steuerstruktur Iteration:

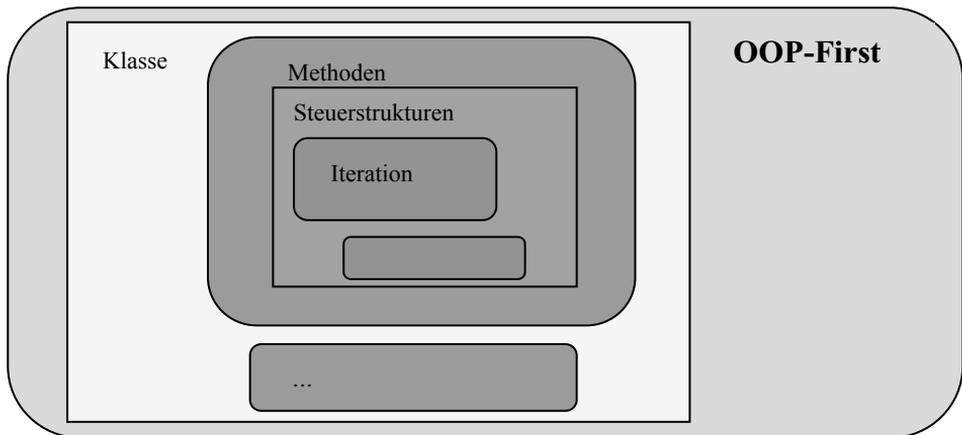


Abbildung 1: Thema Iteration beim OOP-First-Ansatz

Zuerst werden mit Hilfe von Klassen Objekte erzeugt und benutzt. Dabei werden entweder vorgefertigte Klassen benutzt, Klassen implementiert und instanziiert, oder Klassen modelliert, implementiert und instanziiert. Je nach Variante werden also mehr oder weniger stark die im Klassen-Konzept enthaltenen Themen wie z.B. Attribute und Methoden allgemein eingeführt, um dann später diese zu konkretisieren (Methode -> Steuerstruktur -> Sequenz, Selektion, Iteration).

Dagegen wird beim objects-later-Ansatz in der Objektorientierten Programmierung auf vorhandenes Wissen aus der Prozeduralen Programmierung zurück gegriffen:

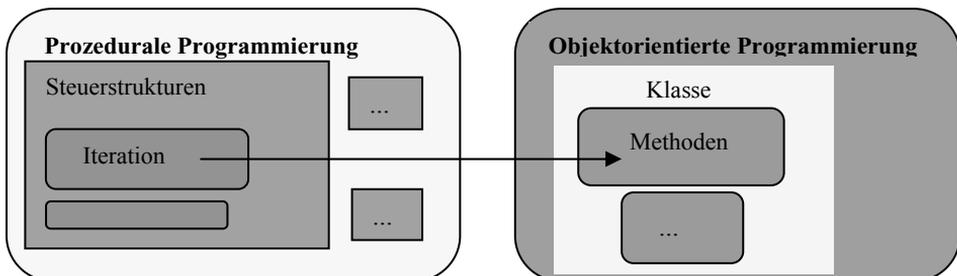


Abbildung 2: Thema Iteration beim OOP-Later-Ansatz

Die Abbildung 2 veranschaulicht, dass beim objects-Later-Ansatz beim Thema „Methoden einer Klasse“ z.B. auf das vorhandene Wissen über die Steuerstruktur Iteration aufgebaut werden kann.

Bei diesem Ansatz gibt es für den Unterrichtenden Freiheitsgrade, es stellt sich nämlich für ihn im Verlauf des Informatik-Unterrichts die Frage: Wann ist der Punkt gekommen, um von der prozeduralen Sichtweise auf die objektorientierte Sichtweise umzusteigen?

Gleichartige Themen bei beiden Ansätzen. Diese Diskussion zeigt, dass der wesentliche Unterschied vom objects-first-Ansatz zum objects-later-Ansatz in der Reihenfolge der Lernthemen – und nicht in den verwendeten Werkzeugen oder Unterrichtsmethoden – zu suchen ist (Wobei natürlich je nach Reihenfolge Werkzeuge anders in den Unterricht eingebettet werden können bzw. müssen).

Daher können für beide Ansätze gleichartige Themen benannt werden [Eh07]. Dazu wurden verschiedene Programmier-Lehrbücher (mit den verschiedensten Ansätzen) im Hinblick auf die einzelnen Programmier-Themen verglichen. Im Ergebnis finden sich immer wieder gleichartige Kapitel bzw. Sequenzen, kleine Unterschiede ausgenommen: Mal firmieren die Steuerstrukturen im Kapitel „Anweisungen“, mal entdeckt man die fundamentalen Datentypen im Unterkapitel „Variablen und Konstanten“ und manchmal findet sich ein Zusatzthema, z.B. „Testen“ (vgl. z.B. [KS07][Er05][Ba08]).

Diese gleichartigen Themen sind:

- Einstieg in die OOP, Klasse und Objekt
- Variablen (bzw. Attribute), Konstanten und fundamentale Datentypen
- Steuerstrukturen: Sequenz, Iteration und Selektion
- Prozeduren (bzw. Operationen, Methoden, Funktionen, Botschaften)
- Komplexe(re) Datentypen
- Vererbung etc.
- Assoziation etc.

Obwohl beide Wege auf den ersten Blick sehr unterschiedlich sind, enthalten sie doch im Endeffekt die gleichen Themen, z.B. die Implementierung von Klassen und die Erzeugung von Objekten, genauso wie die Themen Daten- und Steuerstrukturen. Nur die Sichtweise ist eine andere:

1. Das objektorientierte Paradigma steht von Anfang an im Mittelpunkt des Anfänger-Programmier-Unterrichts (OOP-First), die Themenfolge ist oft wie folgt:

- Klasse und Objekt
- Attribut (inkl. Datentypen)
- Methode (inkl. Steuerstrukturen)
- Vererbung
- Assoziation

2. Das objektorientierte Paradigma baut auf das prozedurale Paradigma auf (OOP-Later), die Themenfolge ist oft wie folgt:

- Variable, Konstante, einfache Datentypen
- Steuerstrukturen: Sequenz, Selektion, Iteration
- Prozedur (Methode)
- Klasse und Objekt

Die Vorgehensweisen, die auf den ersten Blick so unterschiedlich wirken, sind aber in der Summe ihrer Themen gleich, nur die zeitliche Abfolge der Themen ist eine andere!

OOP-First und OOP-Later. Im Folgenden soll der Begriff **OOP-First** verwendet werden, wenn im Anfangsunterricht gleich mit dem Begriff Objekt bzw. Klasse eingestiegen wird und der Schwerpunkt eher in der Programmierung bzw. Implementierung als in der Modellierung liegt (dies ist auch die Abgrenzung zu objects first, OOM und OO-First). Instruktionspsychologisch lässt sich hier im Sinne einer Top-Down-Argumentation positiv anführen, dass den Lernenden die Ziele und Anwendungsmöglichkeiten schnell deutlich werden, auf die sie hin arbeiten. Dies könnte sich emotional günstig auswirken.

Dagegen wird der Begriff **OOP-Later** gesetzt, wenn ein Teil der prozeduralen Themen vor dem Einstieg in die OO-Themen erfolgt und auch hier wieder der Schwerpunkt eher in der Programmierung bzw. Implementierung liegt [Eh07]. Instruktionspsychologisch könnte man mit dem Vorteil eines systematischen Wissensaufbaus argumentieren.

Mit Hilfe dieser Definition kann nun (unter Fokussierung auf OOP-First und OOP-Later und der dort enthaltenen Programmier-Komponente) der objects-first-Ansatz mit dem objects-later-Ansatz – unter dem als zentral herausgearbeiteten Aspekt der unterschiedlichen Sequenzierung von Lerninhalten – verglichen werden.

3 Empirische Studie zum Vergleich von OOP-First und OOP-Later

In diesem Abschnitt werden ausgewählte Aspekte einer größeren empirischen Studie vorgestellt und diskutiert. Wir konzentrieren uns auf den Vergleich in den Lernergebnissen. Die zentrale Forschungsfrage lautet: Gibt es nach einem Jahr (signifikante) Unterschiede im Lernerfolg der Schülerinnen und Schüler der beiden Klassen? Kann man diese aufschlüsseln nach einzelnen Themengebieten?

Design der Studie. Zwei parallele Schulklassen wurden ein Jahr lang in Informatik beschult. Die Themen waren gleich, die Themenreihenfolge war aber unterschiedlich. Das zugrunde liegende Forschungsdesign der empirischen Studie ist ein kausales, welches zum Ziel hat, Zusammenhänge zwischen Variablen aufzudecken. Dazu wurde ein Experiment durchgeführt, indem sich die Eingangsvariablen (fast) nur durch die unterschiedliche Reihenfolge der Themen unterschieden. Über eine geschichtete Stichprobenauswahl wurden einzelne Schüler-Merkmale (wie Geschlecht, Alter, etc.) schon von vornherein in ihrem richtigen Verhältnis im Sample repräsentiert. Für jedes Thema wurden Fachkompetenzen formuliert und daraus Aufgaben abgeleitet, die als Indikatoren für das Erreichen der einzelnen Kompetenz dienen. Die Programmiersprache war durch schulinterne Vorgaben auf Java festgelegt, der OOP-Einstieg (ein Thema von neun Themen) erfolgte mit BlueJ [BK03], alle Programme für die anderen Themen wurden mit dem JavaEditor [Rö09] erstellt. Durch ständige Absprachen der beiden Unterrichtenden wurde dafür gesorgt, dass die Themen gleichartig und gleichlang unterrichtet wurden. Die Gleichartigkeit der Klassen (im Hinblick auf das mathematische Verständnis) wurde durch den selben Mathematik-Lehrer in beiden Klassen überprüft. In einer Vorstudie (Schuljahr 2006/2007) wurde das „Setting“ der Studie erprobt, damit in der Hauptstudie tatsächlich bis auf die Themenreihenfolge alle anderen Variablen möglichst gleichartig sind.

Der Lernerfolg wurde am Ende des Schuljahres mittels eines Tests (Post-Test bzw. Vergleichstest) über ca. 120 Minuten ermittelt: OOP-abhängige und OOP-unabhängige Themen waren dabei mit ca. 50% vertreten. Für den Vergleich eventueller Unterschiede in der Nachhaltigkeit wurde ein vergleichbarer Test 8 Wochen nach dem letzten Test (also nach den Sommerferien 2008) als Follow-Up-Test bzw. Nachhaltigkeitstest geschrieben.

Die verschiedenen Fragebögen wurden aus den Forschungsfragen entwickelt und überarbeitet bzw. über die Vorstudie evaluiert.

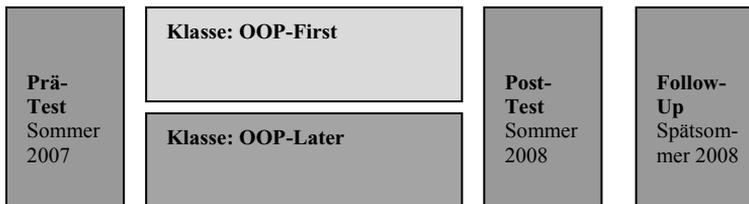


Abbildung 3: Forschungsdesign (Hauptstudie)

Durchführung der Studie. Die Studie wurde mit zwei Schulklassen in der Einführungsphase des Beruflichen Gymnasiums am OSZ Informations- und Medizintechnik (www.oszimt.de) in Berlin durchgeführt. Damit gelten die folgenden Rahmenbedingungen:

- die Schülerinnen und Schüler haben in der Regel keine Programmiererfahrungen und sind eher leistungsschwach,
- die Schülerinnen und Schüler sollen neben der Modellierung auch relativ schnell real programmieren (hier: Java),
- die Schülerinnen und Schüler sollen in einem Schuljahr (à 3h / Woche) von Datentypen, Steuerstrukturen bis zur Vererbung in der OOP sowohl imperative als auch objektorientierte Sichtweisen kennengelernt haben.

Für die Schülerinnen und Schüler ist die Einführungsphase in der gymnasialen Oberstufe das erste Schuljahr (11. Klasse) an der Schule. Daher konnten vorher die Klassen für den Vergleich nach Geschlecht (in der Regel männlich), Alter (ca. 17 Jahre), Vorbildung (in der Regel Realschulabschluss) und Mathematik- und Deutsch-Note gleichartig zusammengesetzt werden.

Die Vorstudie lief im Schuljahr 2006/2007, die Hauptstudie ist nach dem Schuljahr 2007/2008 beendet worden.

Die Themen in beiden Klassen waren gleichartig, nur die Reihenfolge der Themen war eine andere. Die insgesamt neun Themenblöcke wurden aus den übereinstimmenden Themen der Programmier-Fachliteratur gewonnen (s. Seite 4), es kam natürlich noch der Einstieg in die IDE hinzu. Es wird dabei mit Absicht von Themen und nicht von Modulen gesprochen, da die Art und Weise der Vermittlung der einzelnen Themen auch abhängig ist von den vorher unterrichteten Themen.

Die OOP-First-Klasse startete gleich mit den OOP-Fachtermini wie Klasse, Objekt, Attribut und Methode:

Reihenfolge	Thema
1	T1: Einführung in die OOP (mit BlueJ)
2	T2: Einführung in den Java-Editor, Attribute (Variablen), fundamentale Datentypen, Steuerstruktur Sequenz
3	T3: Klasse und Objekt
4/5*	T4: Methoden
5/4*	T5: Steuerstruktur: Selektion (inkl. Struktogramm)
6	T6: Steuerstruktur: Iteration (inkl. Struktogramm)
7	T7: Komplexere Datentypen: Arrays und Strings
8	T8: Vererbung (inkl. abstrakte Klassen und abstrakte Methoden)
9	T9: Assoziationen (inkl. Klassenattribute und -methoden)

Abbildung 4: Reihenfolge der OOP-First-Themen
 (*linke Zahl: Reihenfolge Vorstudie, rechte Zahl: Reihenfolge Hauptstudie)

Die OOP-Later-Klasse fokussierte sich dagegen erst einmal (ca. ½ Schuljahr) auf prozedurale Themen inklusive den Daten- und Steuerstrukturen:

Reihenfolge	Thema
1	T2: Einführung in den Java-Editor, Variablen (Attribute), fundamentale Datentypen, Steuerstruktur Sequenz
2	T5: Steuerstruktur: Selektion (inkl. Struktogramm)
3	T6: Steuerstruktur: Iteration (inkl. Struktogramm)
4	T7: Komplexere Datentypen: Arrays und Strings
6/5*	T1: Einführung in die OOP (mit BlueJ)
7/6*	T3: Klasse und Objekt
5/7*	T4: Methoden
8	T8: Vererbung (inkl. abstrakte Klassen und abstrakte Methoden)
9	T9: Assoziationen (inkl. Klassenattribute und -methoden)

Abbildung 5: Reihenfolge der OOP-Later-Themen
 (*linke Zahl: Reihenfolge Vorstudie, rechte Zahl: Reihenfolge Hauptstudie)

Bei der letzten Darstellung erkennt man schon, dass Lehren aus der Vorstudie gezogen wurden: Das Thema „Prozeduren“ wurde jetzt nicht mehr vor dem OOP-Einstieg als „statische Methoden“, sondern erst nach dem OOP-Einstieg als „Methoden“ unterrichtet.

Prä-, Post- und Follow-Up-Test. Für den Vergleich des Lernerfolgs wurde in beiden Klassen nach einem Jahr der selbe Test (Vergleichstest bzw. Post-Test) über 120 Minuten geschrieben. In der Vorstudie wurden die fünf Themen Daten- und Steuerstrukturen, Struktogramme, Methoden, „statische OOP“, UML und das Zusatzthema „dynamische OOP“ abgefragt. Beim letzten Thema erhielten die Schüler Quelltext eines Anwendungsprogramms, in dem verschiedenen Objekte erzeugt und auf ihnen verschiedene Methoden aufgerufen wurden. Die Schülerinnen und Schüler sollten dann die dynamische Entwicklung der Objektzustände nachvollziehen können. In der Hauptstudie wurden alle neun unterrichteten Themen einzeln abgefragt inklusive zweier Zusatzthemen („dynamische OOP“ und OOM: aus einer Problembeschreibung sollten Klassen modelliert werden). OOP-abhängige (UML, Klasse, Vererbung etc.) und OOP-unabhängige Themen (Datentypen, Steuerstrukturen, Struktogramme etc.) waren dabei mit jeweils ca. 50% vertreten. Maximal 30% waren Multiple-Choice-Aufgaben.

4 Ergebnisse

Vorstudie. Bei einem ersten Probedurchlauf im Schuljahr 2006/2007 waren die objektiven Lernergebnisse (ermittelt durch den Post-Vergleichstest am Ende des Schuljahrs) bei der OOP-Later-Klasse teilweise signifikant besser, gerade im Hinblick auf die OOP-Themen! [ES07]

Thema	Punkte	OOP-First	OOP-Later
Daten- und Steuerstrukturen	20	61%	59%
Methoden	20	67%	64%
Struktogramme	20	60%	64%
„Statische OOP“	20	72%	79%
UML-Klassendiagramme und -Objektdiagramme	20	72%*	85%*
Zusatz: „Dynamische OOP“	10	50%*	67%*
Gesamt	110	65%	70%

Abbildung 6: Ergebnisse des Vergleichstests (Post-Test) in der Vorstudie 2006/2007
(*signifikante Ergebnis-Unterschiede sind fett gedruckt)

Auffällig war auch, dass die OOP-First-Klasse mit den Themen schneller vorangekommen ist, so dass ein kurzes Projekt das Schuljahr abschloss. Dies kann ein Vorteil des OOP-First-Wegs sein, evt. aber auch die Erklärung dafür, warum die Schüler schlechter abgeschnitten haben. Vielleicht „verführt“ der OOP-First-Einstieg zu einem schnelleren Vorgehen. Eine andere Erklärung ist, dass Schüler das zuletzt Gelernte am Besten abrufen können. Dies waren bei der OOP-Later-Klasse ja die OOP-Themen!

Hauptstudie. In der Hauptstudie 2007/2008 wurde der Vergleichstest umgestaltet. Die einzelnen unterrichteten Themengebiete finden sich jetzt besser wieder:

Thema	Punkte	OOP-First	OOP-Later
T2: Datentypen und Steuerstruktur Sequenz	10	6,8	7,0
T5: Steuerstruktur Selektion	10	6,3	7,1
T6: Steuerstruktur Iteration	10	5,8	6,0
T7: Arrays und Strings	10	5,7*	3,7*
T4: Methoden	10	6,1	4,8
T1/T3: Einführung in die OOP / Klasse und Objekt	20	14,6	14,9
T8: Vererbung	10	7,6	6,9
T9: Assoziation	10	6,2*	4,9*
Zusatz: Dynamische OOP	10	7,1	7,0
Zusatz: Objektorientierte Modellierung (OOM)	10	6,7	5,9
Gesamt	110	72,9	68,2
<i>Mathematik-Note (Punkte)</i>		8,8	7,8

Abbildung 7: Ergebnisse des Vergleichstests (Post-Test) in der Hauptstudie 2007/2008
(*signifikante Ergebnis-Unterschiede sind fett gedruckt)

Das etwas bessere Gesamt-Abschneiden der OOP-First-Klasse könnte einfach durch die besseren Mathematik-Leistungen dieser Klasse erklärt werden, eine Korrelation zwischen Mathematik- und Informatik-Note voraussetzend. Der Unterschied bei dem Thema „Methoden“ ist zwar auffällig, aber nur die Ergebnisse bei den Themen „Arrays

und Strings“ und „Assoziation“ sind signifikant unterschiedlich. Vor einer Bewertung dieser Leistungs-Unterschiede sollen aber erst einmal die Ergebnisse des Nachhaltigkeits- bzw. Follow-Up-Tests (ca. 8 Wochen später) dargestellt werden:

Thema	Punkte	OOP-First	OOP-Later
T2: Datentypen und Steuerstruktur Sequenz	10	6,4	7,2
T5: Steuerstruktur Selektion	10	6,5	7,4
T6: Steuerstruktur Iteration	10	5,7	5,4
T7: Arrays und Strings	10	3,6	3,8
T4: Methoden	10	5,3	4,6
T1/T3: Einführung in die OOP / Klasse und Objekt	20	13,4	13,0
T8: Vererbung	10	5,8	6,0
T9: Assoziation	10	2,5	1,6
Zusatz: Dynamische OOP	10	5,3	5,1
Zusatz: Objektorientierte Modellierung (OOM)	10	5,3	6,3
Gesamt	110	59,8	60,4

Abbildung 8: Ergebnisse des Nachhaltigkeitstests (Follow-Up-Test) in der Hauptstudie 2007/2008 (Es gibt keine signifikanten Ergebnis-Unterschiede)

Hier fallen die deutlichen Einbrüche der Ergebnisse der OOP-First-Klasse auf, sowohl im Endergebnis als speziell auch bei den vorher signifikant unterschiedlichen Themen-Ergebnissen. Bei der Ursachenforschung wurde schnell klar: Beim Vergleichstest waren die Klassen von den beiden beteiligten Informatik-Lehrern sehr unterschiedlich vorbereitet worden! Daher sind die deutlich objektiveren Ergebnisse jene des Nachhaltigkeitstest (auf den überhaupt nicht vorbereitet wurde) und sollen deshalb im Folgenden diskutiert werden.

Ergebnis1. Es gibt keine Unterschiede zwischen dem OOP-First- und dem OOP-Later-Vorgehen bei der Nachhaltigkeit! Sowohl was das Gesamtergebnis als auch was die einzelnen Themen anbelangt, ergeben sich keine signifikanten Unterschiede. Beide Vorgehensweisen produzieren auf lange Sicht vergleichbare Lernerfolge.

Die Ergebnisse für beide Klassen können gemittelt, quantitativ geordnet und dann in drei Leistungsbereiche aufgeteilt werden:

Thema	Punkte	Beide Klassen
T5: Steuerstruktur Selektion	10	7,0
T2: Datentypen und Steuerstruktur Sequenz	10	6,8
T1/T3: Einführung in die OOP / Klasse und Objekt	20 (10)	13,2 (6,6)
T8: Vererbung	10	5,9
Zusatz: Objektorientierte Modellierung (OOM)	10	5,8
T6: Steuerstruktur Iteration	10	5,6
Zusatz: Dynamische OOP	10	5,2
T4: Methoden	10	5,0
T7: Arrays und Strings	10	3,7
T9: Assoziation	10	2,1
Gesamt	110	60,3

Abbildung 9: Geordnete Ergebnisse des Nachhaltigkeitstests in der Hauptstudie 2007/2008

Wenn man die Ergebnisse im Hinblick auf OOP-abhängige und OOP-unabhängige Themen betrachtet, fällt auf, dass sich die beiden Themenarten in allen drei Leistungsbereichen wiederfinden.

Ergebnis2. Die Themen der OOP sind nicht schwerer als die prozeduralen Themen! In jeder Themenart finden sich leichtere und schwerere Themen wieder.

Der Follow-Up-Test bzw. Nachhaltigkeitstest war nicht identisch mit dem Post-Test, aber Aufgabe für Aufgabe vergleichbar. Der Prä-Test war ein stark verkürzter Post-Test und dient nur dem Nachweis, dass die beiden Klassen auf dem gleichen (tiefen) Ausgangsniveau starten. Da im Prä-Test nur die ersten 8 Themen abgefragt wurden, finden sich im folgenden Diagramm auch beim Post- und beim Follow-Up-Test nur die ersten 8 Themen wieder:

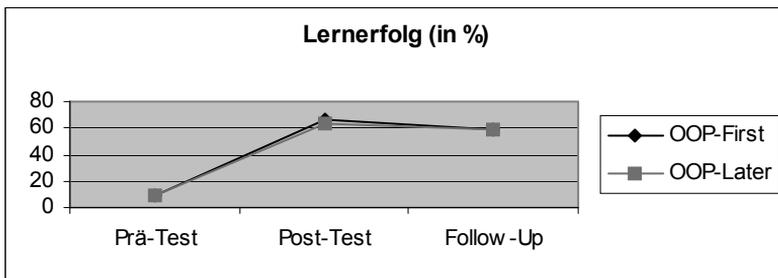


Abbildung 10: Lernerfolg in der Hauptstudie 2007/2008

(Bezogen auf die ersten 8 Themengebiete, Prä-Test erfolgte im nächsten Jahrgang parallel zum Follow-Up-Test)

Die Schülerinnen und Schüler kommen fast ohne Programmier-Vorkenntnisse in die Einführungsphase (9,8% richtige Antworten sind eher den Multiple-Choice-Aufgaben geschuldet). Nach einem Schuljahr erreichen sie ca. 64,6% des erwünschten Wissens, durch die Sommerferien vergessen sie (nur) ca. 5,8%!

Ergebnis3. Die Unterschiede in den Lernerfolgskurven sind nicht signifikant. Vom Gesamtergebnis her gesehen gibt es keine Unterschiede beim Post-Test und Follow-Up-Test (bei vergleichbarem Prä-Test) zwischen dem OOP-First- und dem OOP-Later-Vorgehen.

5 Interpretation, neue Forschungsfragen und Zusammenfassung

Interpretation. Das letzte Ergebnis könnte so interpretiert werden, dass, wenn man sich für den Informatik-Anfängerunterricht inkl. OOP-Einstieg nur genug Zeit nimmt (in diesem Fall ca. 100 Stunden), es unabhängig von den Lernstilen, den Lernmethoden, den Lerntools, der verwendeten Programmiersprache etc. den gleichen Lernerfolg gibt.

Auch wenn das zentrale Ergebnis ist, dass sich die Lernergebnisse mittelfristig nicht unterscheiden, so wird es doch bestimmt „lokale“ Unterschiede im Lernprozess geben. Dazu könnte nach jedem Thema eine Momentaufnahme gemacht werden:

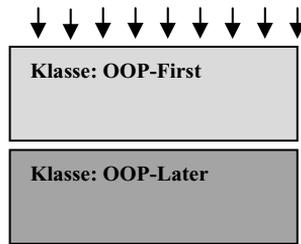


Abbildung 11: Mögliche Momentaufnahmen nach jedem Thema

Durch diese Prozessbeobachtung könnte sich dann herausstellen, dass jedes Vorgehen spezifische Vor- und Nachteile hat, z.B. bei den Themenübergängen oder bei der von den Schülern erlebten „Themenschwere“ (siehe dazu spätere Veröffentlichung).

Die Themen „Assoziation“ und „Arrays und Strings“ hatten den wenigsten Lernerfolg. Die Schwere dieser Themen könnte darin begründet sein, dass beim Thema „Arrays“ es zu einer Strukturierung eines Datentypen kommt (z.B. von int auf int-Feld) inkl. der Anwendung eines Algorithmus’ (sprich: Iteration, um die Feldstruktur zu durchlaufen). Wenn eine 1-n-Assoziation mit Hilfe eines Arrays implementiert wird, dann ist es logisch, dass das Thema „Assoziation“ von den Schülern auch als komplexer erlebt wird.

Die empirische Studie bestätigt (quasi nebenbei) auch noch einmal den Zusammenhang bzw. Unterschied von OO-first und OO-later: Es konnten neun Themengebiete definiert werden, die sowohl beim OOP-First-Ansatz als auch beim OOP-Later-Ansatz zur Geltung kamen. Nach einem Jahr sind in beiden Varianten alle 9 Themengebiete unterrichtet worden, nur die zeitliche Reihenfolge der Themen hat variiert.

Einschränkungen der Gültigkeit. Mögliche Fehl- oder Überinterpretationen sind wie folgt möglich:

- Nicht entschieden werden kann über den Einfluss der gesetzten Randbedingungen, z.B. bei Wahl anderer Werkzeuge, Programmiersprachen etc. Trotzdem ist es schon ein Erfolg, einen systematischen Vergleich geschafft zu haben. Darauf können Folgeuntersuchungen aufbauen.
- Gegebenenfalls gibt es „lokale“ Unterschiede im Prozess (siehe oben). Auch ist bisher nur die Schülersamtheit betrachtet worden. Vielleicht gibt es aber in Teilgruppen (z.B. sehr gute oder sehr schlechte Schüler) signifikante Unterschiede beim Lernerfolg in Abhängigkeit vom Vorgehen.
- Die Studie bezieht sich auf OOP nicht auf objects first, wobei der Unterschied marginal sein mag (Hinweis: Auch bei der zusätzlichen Modellieraufgabe gab es keinen Unterschied). Größere Unterschiede könnten sich ergeben, wenn OOP-First mit OOP-Later verglichen wird. Dies ist dann aber eher eine Frage der Gewichtung von Lernzielen als eine didaktisch-methodische Frage.
- Die Testfragen waren überwiegend (79%) aus dem Anforderungsbereich II, der Anforderungsbereich III fehlte gänzlich. Quelltext-Entwicklung erfolgte auf dem Papier, nicht am Rechner.
- Die Probandenzahl der beiden Schulklassen betrug 40 Schülerinnen und Schüler.

Neue Fragestellungen. Aus der Studie lassen sich weitere Fragen entwickeln, z.B.:

- Was passiert, wenn einzelne Variablen geändert werden, z.B. die Programmiersprache (Java) oder das OOP-Einstiegstool (BlueJ)?
- Was passiert, wenn das sequentielle OOP-First-Vorgehen auf ein spiralförmiges OOP-First-Vorgehen umgestellt wird?

Zusammenfassung. Die Fragestellung braucht in Zukunft nicht mehr zu lauten, mit welchem Paradigma die Lehrerin bzw. der Lehrer in den Informatik-Anfängerunterricht einsteigt, da bezogen auf ein Schuljahr (mit ca. 100-120h) die Klassen fast das Gleiche lernen. Der Unterrichtende muss sich vielmehr Gedanken machen, warum bestimmte Themen aus welchen Gründen von den Schülern als schwer erlebt werden bzw. mangelhaft gelernt werden und warum bestimmte Themenübergänge Schwierigkeiten bereiten. Auf den Informatik-Unterricht des OSZ IMT bezogen könnte man z.B. das Thema „Assoziationen“ im ersten Schuljahr gänzlich herausnehmen und mit der gewonnenen Zeit das vertiefende (sprich: längere) Unterrichten von „Arrays und Strings“ bewirken.

Literaturverzeichnis

- [Ba08] Balzert, Helmut: Java, der Einstieg in die Programmierung
W3L GmbH 2008, 2. Auflage, ISBN 978-3-868-34000-6
- [BK03] Barnes D.J., Kölling M.: Java lernen mit BlueJ
Pearson Education Deutschland, 2003, 3. Auflage, ISBN 0-13-197-629X
- [BS07] Bennedsen, Jens; Schulte, Carsten: What does “Objects-First” Mean? An International Study of Teachers’ Perceptions of Objects-First. In: Lister, R. und Simon, Hrsg. (2007), Koli Calling 2007, Finland, ACS., <http://crpit.com/confpapers/CRPITV88Bennedsen.pdf>
- [De03] Decker, A.: A tale of two paradigms.
J. Comput. Small Coll. 19, 2 (Dec. 2003), 238-246.
- [Di07] Diethelm, Ira: Strictly models and objects first, Dissertation
Kassel, Universität, 2007, urn:nbn:de:hebis: 34-2007101119340
- [Eh07] Ehlert, Albrecht: Studie: Objects-First- und Objects-Later-Einstieg
Praxisband der 12. Fachtagung "Informatik und Schule - INFOS 2007", Seite 17 – 20,
Herausgeber: Stechert P., Universität Siegen 2007, ISBN 978-3-936533-23-1
- [Er05] Erlenkötter, Helmut: C++ - Objektorientiertes Programmieren von Anfang an
Rowohlt Tb.; Auflage: Erw. N.-A. (Mai 2005), ISBN 978-3499600777
- [ES07] Ehlert A., Schulte C.: Learners Views on Objects-First and Objects-Later -
Results of an Exploratory Study, Report on the 11th Workshop TLOOC at ECOOP 2007
Herausgeber: Börstler J., Hadar I., ISBN 978-3-540-78194-3
- [KS07] Krüger Guido, Stark Thomas: Handbuch der Java-Programmierung
Addison-Wesley, München; November 2007, ISBN 978-3827323736
- [Li06] Lister, R., Berglund, A., Clear, T., Bergin, J., Garvin-Doxas, K., Hanks, B., Hitchner, L.,
Luxton-Reilly, A., Sanders, K., Schulte, C., and Whalley, J. L. 2006. Research
perspectives on the objects-early debate. SIGCSE Bull. 38, 4 (Dec. 2006), 146-165
- [Re06] Reges, S. 2006. Back to basics in CS1 and CS2. In Proceedings of the 37th SIGCSE,
SIGCSE '06. ACM, New York, NY, 293-297
- [Rö09] Röhner, Gerhard: JavaEditor, '09, <http://lernen.bildung.hessen.de/informatik/javaeditor/>
- [Sp05] Spolwig, Siegfried.: Karel D. Robot – Der Delphi Karel, OSZ Handel, Berlin, 2005
http://www.oszhandel.de/gymnasium/faecher/informatik/delphi_karel/index.htm (3.5.09)

Kompetenzentwicklung mit Informatiksystemen – KIS

Christian Kollee, Peer Stechert, Sigrid Schubert

Didaktik der Informatik und E-Learning
Universität Siegen
Hölderlinstraße 3
57076 Siegen

{christian.kollee|peer.stechert|sigrid.schubert}@uni-siegen.de

Abstract: Im Herbst 2008 wurde das Unterrichtsprojekt KIS durchgeführt. Frühere Forschungen zeigen, dass der Einsatz von Strukturmodellen einen wesentlichen Beitrag zu einer Kompetenzentwicklung mit Informatiksystemen liefern kann. Mit Unterrichtsbeobachtungen wurden Erkenntnisse zu Lehr-Lern-Prozessen im Unterrichtsprojekt KIS gewonnen. Zu dem Von-Neumann-Blockmodell und den Schichtenmodellen wurden kognitiven Barrieren bei den Schülern festgestellt. Dieser Artikel diskutiert zwei dieser Barrieren genauer: Abstraktionsstufen der verwendeten Strukturmodelle und die entsprechende Fachsprache. Daraus werden wissenschaftliche Fragestellungen für die weitere Kooperation zwischen Schule und Universität abgeleitet.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21.–24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Informatische Bildung mit Mobiltelefonen? Ein Forschungsbericht

Matthias Heming
matthias@familie-heming.de

Abstract: Die Bildungsstandards für Informatik sind nicht mit dem Ziel geschaffen worden, dass nach ihnen Informatikunterricht gestaltet werden kann, im Gegenteil, die Standards »müssen [...] im Schulalltag noch mit Leben gefüllt werden« bzw. »Unterrichtssequenzen in Bezug zu den Standards gesetzt werden«, um »Beispiele guten, an Bildungsstandards orientierten Unterrichts [zu sammeln]« [GI08, S. VII]. Dieser Beitrag zeigt auf, dass basierend auf den Grundsätzen der Chancengleichheit und in Bezug zu gegenwärtigen und zukünftigen Lebenssituationen der Schülerinnen und Schüler, »nicht nur die Informatiksysteme *PC* und *Internet* in den Blick genommen werden [müssen]« [GI08, S. 5], sondern auch und gerade mit Mobiltelefonen standardkonformer Unterricht gestaltet werden kann.

An der Willy-Brandt-Gesamtschule Bergkamen wurde im Schuljahr 2007/2008 ein bisher wohl einzigartiges Projekt gestartet. Seit diesem Schuljahr arbeitet ein kompletter Informatikgrundkurs in der Oberstufe durchgängig mit Mobiltelefonen als Informatiksystem. Der Erfolg dieses Projektes wird anhand erster Interviews näher untersucht und gleichzeitig werden Beispiele vorgestellt, die dazu anregen sollen, selbst einmal ein solches Experiment durchzuführen. Einer dieser Vorschläge stellt mit PyObjVG eine neue Erweiterung der Stifte-und-Mäuse-Bibliothek vor, welche einen einfachen Zugang zur objektorientierten Programmierung anhand von Visualisierungen von Vektorgrafiken ermöglicht.

1 Mobiltelefon vs. Computer – Ein erster Vergleich

In Form eines Werkzeuges wird der Computer in vielen Fächern genutzt, z. B. zur Erstellung digitaler Präsentationen. Die Nutzung von Textverarbeitungsprogrammen in Deutsch oder Englisch wird ergänzt mit der Nutzung von Tabellenkalkulationen zur Visualisierung von Zusammenhängen in der Mathematik oder den Naturwissenschaften. Zusammen mit dem Internet als *das neue Medium* kann man diesen Einsatz durchaus kritisch betrachten, doch in diesem Zusammenhang viel wichtiger ist die Frage, mit welchem Ziel der Computer in der Informatik benutzt wird. Im ersten Moment scheint die Frage leicht zu beantworten, daher soll sie etwas präzisiert und in der Perspektive leicht verändert werden.

Für welche Zwecke ist der Einsatz eines Computers im Informatikunterricht zwingend notwendig und kann nicht durch ein anderes Informatiksystem ersetzt werden?

Diese Frage ist wesentlich schwerer zu beantworten, betrachtet man die Konsequenzen, die der Computereinsatz mit sich bringt, so wird einem deutlich bewusst, dass die Suche

nach einer Antwort für die unterrichtliche Praxis nicht unwichtig ist.

Der wohl wichtigste Punkt sind die Kosten in der Anschaffung und Wartung von stationären oder mobilen Computern. Da diese nicht unerheblich sind, erscheint es selbstverständlich, dass nicht jeder Schüler bzw. jede Schülerin mit einem solchen Gerät ausgestattet werden kann, sondern nur innerhalb speziell ausgestatteter Räume (z. B. Computerräume) mit diesen arbeiten kann. Bei stationären Computern sind räumliche Gegebenheiten festgelegt. Der typische Computerraum ist häufig nicht auf die didaktischen Bedürfnisse des Unterrichts ausgelegt. Aus technischen Gründen können die Schülerarbeitsplätze nicht verrückt bzw. verstellt werden, die Arbeitsplätze können wenig individuell gestaltet werden. Dadurch sind Gruppenarbeiten häufig nicht realisierbar, die am häufigsten verwendete Arbeitsform ist die der Partnerarbeit, wobei diese bereits durch die zu geringe Anzahl der vorhandenen Geräte forciert wird. Beim Frontalunterricht stellen Bildschirme eine Sichtbarriere zwischen Lehrkraft und Schülerinnen und Schülern dar, welche in Verbindung mit Maus, Tastatur und eventuell sogar dem Computergehäuse einen Gutteil der Tischfläche des Arbeitsplatzes einnehmen.

Auch mobile Geräte stellen nur begrenzt eine Alternative dar. Bei ihnen können die Räumlichkeiten zwar wesentlich flexibler gestaltet werden, was bleibt, sind jedoch Probleme an anderer Stelle. Es muss dasselbe Gerät von verschiedenen Lernenden benutzt werden. Dabei erscheint es selbstverständlich, dass durch eine entsprechende Rechtevergabe eventuelle Fehlkonfigurationen vermieden werden müssen, damit sich diese nicht zum Nachteil für die im folgenden am Gerät arbeitenden Schüler und Schülerinnen auswirken. Diese im Schulbetrieb durchaus notwendige Einschränkung¹ steht jedoch im deutlichen Gegensatz zu Notwendigkeiten des Informatikunterrichts, der Schüler und Schülerinnen Möglichkeiten geben muss, ein gegebenes Informatiksystem in allen Einzelheiten zu erforschen, es selbstständig zu erweitern, seine Grenzen zu testen.

Das Mobiltelefon kann bei dieser Argumentation an entscheidender Stelle punkten. Unter zugegebenermaßen positiven Bedingungen müssen von schulischer Seite keinerlei Kosten getragen werden, da die Mobiltelefone von Schülern und Schülerinnen selbst mit in den Unterricht gebracht werden. Dies ist durchaus nicht als utopisch anzusehen, bereits die Daten der JIM-Studie 2007 sprechen für sich: »Bei den ab 14-Jährigen kann man von einer Handy-Vollversorgung sprechen« [MPF07, S. 55]. Die Studie des letzten Jahres spricht dieses noch deutlicher aus:

Ein Mobiltelefon zu haben ist für Jugendliche nun schon seit einigen Jahren eine absolute Selbstverständlichkeit – inzwischen besitzen 95 Prozent ein eigenes Handy, unabhängig von Geschlecht oder Bildungshintergrund [MPF08, S. 59].

Dahinter bleibt der Computer mit einer Quote von ca. 70 Prozent eindeutig zurück.

Auch wenn bei vielen gerade älteren Menschen das Mobiltelefon tatsächlich nur zum Telefonieren verwendet wird, so spricht die JIM-Studie 2008 im weiteren das aus, was jedem

¹Dabei muss erwähnt werden, dass weniger starke Einschränkungen notwendig sind, wenn Schülerinnen und Schüler sich im Sinne einer Netiquette bewusst sind, welche Konsequenzen ihr Verhalten hat und dementsprechend ein verantwortungsvoller Umgang mit den Systemen möglich ist. Dies ist erklärtermaßen ein Ziel des Informatikunterrichts.

Informatiker sofort klar ist: Das heutige Mobiltelefon ist in allen Belangen ein vollständiges Informatiksystem.

Die Handys, die Jugendliche heute zur Verfügung haben, sind mit früheren Geräten kaum vergleichbar. Neun von zehn Handys haben eine Digitalkamera integriert, etwa vier Fünftel der Jugendlichen können mit dem Handy unterwegs ins Internet gehen oder Daten über Bluetooth tauschen. Drei Viertel können mit dem MP3-Player des Mobiltelefons auch Musik abspielen und 61 Prozent können Radioprogramme empfangen [MPF08, S. 61].

Neben den veränderten Besitzverhältnissen sprechen jedoch auch andere Faktoren für den Einsatz im Unterricht. Auch wenn im ersten Moment für einen computergewöhnten Erwachsenen abschreckend, ist gerade die Größe des Geräts einer dieser wesentlichen Faktoren. Für Mobiltelefone muss kein Platz auf dem Tisch gemacht werden und falls sie in einer konkreten Unterrichtssituation stören, so können sie in die Hosen-, Jacken- oder Schultasche gesteckt werden. Eventuelle Probleme in der Bedienbarkeit² können dafür Sorge tragen, dass nicht die Bedienung eines Informatiksystems, sondern dessen Struktur und Aufbau im Vordergrund stehen.

Das Mobiltelefon nimmt außerdem eine besondere Position ein, da es sowohl im privaten Bereich, als dann auch in der Schule genutzt wird. Durch die private Nutzung haben Schülerinnen und Schüler Interesse daran, die Funktionalität des Systems aufrecht zu erhalten, die Lernenden übernehmen die verantwortungsvolle Rolle des Systemadministrators. Als positiver Nebeneffekt sind finanzielle Belastungen, welche durch zusätzliche Wartungsarbeiten auf schulischer Seite entstanden wären, reduziert.

Als Argument zur Chancengleichheit betrachte man die bereits angesprochenen Zahlen von 95% Gerätebesitz bei Mobiltelefonen und 70% bei Computern unter dem geschlechtsspezifischen Aspekt. 96% aller Mädchen besitzen ein eigenes Gerät und damit sogar ein wenig mehr die Jungen (94%) während mit 64% zu 77% deutlich mehr Jungen einen Computer ihr Eigen nennen können [MPF08, S. 10]. Betrachtet man die Rolle verschiedener Medien im Alltag der Jugendlichen ebenfalls unter dem geschlechtsspezifischen Aspekt, so liegen Jungen in computerlastigen Bereichen teilweise leicht (Computer: 87% zu 91%, Internet: 83% zu 85%, Tageszeitung online: 10% zu 14%, Zeitschriften online: 7% zu 14%) aber auch sehr stark (Computerspiele: 13% zu 47%) vorne [MPF08, S. 13]. Mit welchem Medium beschäftigen sich mehr Mädchen? Mit Mobiltelefonen, und das sogar mit einem Vorsprung von mehr als zehn Prozentpunkten (91% zu 78%).

Nicht nur die durchgeführten Interviews zeigen, dass die Sichtweise auf das Fach Informatik stark in Verbindung steht mit der Nutzung von Computern. Die Nutzung von Computern im Informatikunterricht muss damit aus der Genderperspektive heraus in Frage gestellt werden. Mit Mobiltelefonen existiert eine Alternative dessen Besitzverhältnisse »unabhängig von Geschlecht oder Bildungshintergrund« ist [MPF08, S. 59].

²Man nehme als Vergleichsmaßstab nicht unbedingt die eigene Tippgeschwindigkeit, sondern beachte, dass Jugendliche, die täglich mit einem solchen Gerät umgehen, ein nicht zu vernachlässigendes Training erfahren haben.

2 Interviews

Auch wenn mit dem letzten Abschnitt an der hervorgehobenen Stellung des Computers Kritik geübt worden ist, so ist die oben formulierte Frage bisher nicht beantwortet. Für welchen Zweck ist ein Computer zwingend notwendig? Falls es aus Schülersicht einen solchen gibt, so müsste er der Klasse, die im Schuljahr 2007/2008 von Herr Humbert unterrichtet wurde, aufgefallen sein. Im Rahmen dieses Forschungsprojektes wurden an der Schule fünf Interviews geführt. Zwei der Interviews fanden mit Jungen des Experimental-kurses statt, drei weitere mit zwei Jungen und einem Mädchen aus dem Parallelkurs ohne Mobiltelefone.

2.1 Zur Durchführung

Da die Sichtweisen auf Informatik im Allgemeinen sehr verschieden sind, sollte im ersten Teil des Gespräches herausgefunden werden, welche Sichtweise die Interviewpartnerin bzw. der Interviewpartner einnimmt. Eventuell könnten bereits in diesem Teil Einflüsse der verwendeten Informatiksysteme erkannt werden. Im zweiten Abschnitt sollte den Bezug zwischen allgemeinen Vorstellungen und konkreten Unterrichtserfahrungen herstellen während der dritte Teil zeitgleich den Fokus auf das verwendete Informatiksystem richtete.

Aus mangelnder Interviewerfahrung wurde ein sehr schwach standardisierter Leitfaden erstellt, der eine größtmögliche Flexibilität und damit erhoffte Lernmöglichkeiten seitens des Autors bietet. Wenn auch der Lerneffekt groß war, so zeigte die Reflexion der Interviewdurchführung, dass die Gesprächsführung nicht optimal war. Die drei Assoziationen mit Informatik, welche als Gesprächseinstieg gedacht waren, gaben nur wenig Anknüpfungspunkte und es wurden Fragen, die von Schülerinnen und Schülern nicht direkt beantwortet werden konnten, bewusst übergangen, um keine unangenehme Situation aufkommen zu lassen. Obwohl die Schaffung einer positiven Atmosphäre grundsätzlich anzustreben ist, so hätte hier durchaus häufiger nachgehakt werden können, so dass von den eingeplanten 15 Minuten mehr als nur zehn hätten genutzt werden können. Ebenfalls schwierig war die Wahrung von Neutralität. Bewertungen der Gesprächspartner wurden zu häufig kommentiert oder gar bestärkt, so dass die eigentliche Schülermeinung aufgrund einer vorerst oberflächlichen Äußerung eventuell gar nicht formuliert wurde. Ähnlich problematisch war die Formulierung der Fragen, welche durch den Leitfaden nur sehr grob vorgegeben wurden. Ein unnötiger, im Unterbewusstsein gesetzter Zeitdruck führte zu vorschnellen Formulierungen und mehreren Fragen hintereinander, von denen im weiteren Gesprächsverlauf meist nur jeweils eine beantwortet wurde.

Aufgrund der geringen Anzahl und dem ungünstigen Verhältnis von Mädchen und Jungen kann basierend auf diesen Interviews leider keine Aussage zur interessanten Thematik der Genderproblematik gemacht werden. Im Rahmen von zukünftigen Untersuchungen soll jedoch in dieser Richtung weiter geforscht werden.

2.2 Zusammenfassung der Analysen

Aufgrund des eingeschränkten Umfangs dieses Beitrags wird hier nicht detailliert auf jedes Interview eingegangen, sondern es werden die aus den Interviews resultierenden Schlussfolgerungen dargestellt. Die Namen der interviewten Personen wurden dabei anonymisiert.

Bei den Befragten, die ohne Mobiltelefone unterrichtet wurden, wird der Computer als wesentlicher Bestandteil der Informatik angesehen. Bei Mario entspricht dieses gerade seinen Berufswünschen, passenderweise hat er »schon ziemlich gute Grundlagen beim Rechner«. Für Personen, die wenig bis kein Interesse im Umgang mit dem Computer haben und lieber »mit Freunden unterwegs sind«, sei Informatik kein passendes Fach. An dieser Stelle beschreibt Mario genau die Situation, in der sich sein Mitschüler David sieht. Die Abhängigkeit des Informatikers vom Computer steht für David im Gegensatz zur Vorstellung seiner eigenen Lebenspraxis als Mensch, »der sehr oft draußen ist, der sehr viel Sport macht«. Isabelle nimmt im Vergleich zu den beiden bereits Erwähnten eine mittlere Position ein. Der Computer ist für sie ein Werkzeug zur Kommunikation (Internet, E-Mails), dessen Funktionsweisen durchaus hinterfragt werden sollten, problematisch ist für sie der Programmieranteil, da sie in dieser Thematik eine starke Hilflosigkeit empfindet.

Es stellt sich nun die Frage, auf welche Weise ein Mobiltelefon im Unterricht die Meinung der Schülerin bzw. der Schüler beeinflussen könnte. Für Mario ist der Einsatz der Geräte weniger interessant, er sieht insbesondere in der Bedienung Schwierigkeiten. Dies ist nicht weiter verwunderlich, wenn man betrachtet, dass die erwähnten »guten Grundlagen beim Rechner« beim Einsatz von Mobiltelefonen nicht mehr verwendbar wären. Die Bedienung ist auch für Isabelle das einzige Problem, da sie sonst keine Unterschiede beim Unterricht mit verschiedenen Informatiksystemen vermutet. David sieht den Einsatz von Telefonen als durchweg positiv, welches in passendem Zusammenhang zu seiner Abneigung gegenüber Computern steht. Seine Freizeitbeschäftigung würde nicht mehr im Gegensatz zu den stationären Geräten stehen, Informatik könnte sich in seinen Alltag integrieren.

Eine ähnliche Bandbreite der Meinungen lässt sich ebenfalls bei den Teilnehmern des Unterrichts mit Mobiltelefonen erkennen. Carsten ist durchweg von diesem Projekt begeistert. Zwar entkräftet er das Argument der etwas schwerfälligeren Bedienung nicht, die positiven Argumente sind jedoch klar überwiegend. Dazu zählt er zum einen die bereits bei David angesprochenen Mobilität, die Unabhängigkeit vom Computerraum, zum anderen das Kennenlernen eines neuen Informatiksystems, er könne nun »Daten [...] verschicken, die man eigentlich nicht verschicken kann«, ein neues Machtgefühl. Patrick ist nicht so positiv eingestellt. Er gibt zwar an, dass die Begeisterung zu Beginn sehr groß war, diese sich jedoch hauptsächlich auf das zur Verfügung Stellen der Geräte bezog. Der Unterricht mit Computern in früheren Schuljahren sei wesentlich besser gewesen. Er hätte sich gewünscht – und das ist besonders zu beachten – ein Spiel wie *Snake* auf den Telefonen zu programmieren.

Betrachtet man die potenziell schwierigere Bedienbarkeit, so muss Unterricht mit Mobiltelefonen genau darauf ausgerichtet werden. Es dürfen keine langen Quelltexte geschrieben werden, sondern es muss sich bei der Programmierung auf das Wesentliche beschränkt werden. Dies könnte positive Auswirkungen auf das Verständnis von Isabelle haben, da

der von ihr als negativ empfundene Programmieranteil reduziert wird bzw. die erzeugten Programme besser zu überblicken sind. Bezogen auf die Kommunikationsmöglichkeiten, die Isabelle in Computern sieht (Internet und E-Mails), ist das Mobiltelefon als mindestens gleichwertig im Vergleich mit dem Computer anzusehen. Mit dem Hinweis auf die Programmierung von einem Spiel wie Snake spricht Patrick einen wichtigen Punkt an. Der Austausch des Informatiksystems alleine bringt keine positiven Konsequenzen mit sich, es ist weiterhin notwendig, auf die Bedürfnisse der Schülerinnen und Schüler einzugehen, um eine hohe Motivation zu erzeugen bzw. zu erhalten.

3 Mobiltelefone im Unterricht

Die folgenden Vorschläge lassen sich grob in zwei unterschiedliche Kategorien einteilen. Dies ist zum Einen die hier *passiv* genannte Nutzung der Mobiltelefone als Anschauungsobjekt. Dabei wird das Gerät zwar praktisch verwendet, es wird jedoch keine zusätzliche Funktionalität benötigt, da nicht programmiert wird, sondern es nur im Rahmen analytischer Zwecke im Vordergrund steht. Solange keine speziellen Python-Programme wie PyObjVG verwendet werden, gehen damit auch keinerlei Einschränkungen auf bestimmte Betriebssysteme bzw. Herstellerfirmen einher. Auf der anderen Seite steht die *aktiv* genannte Nutzung, welche sich auf das eigenständige Implementieren von Programmen bezieht.

Wenn auch im bisherigen Text bewusst vermieden, so muss an dieser Stelle darauf eingegangen werden, dass längst nicht alle Mobiltelefone die Programmierung auf dem Gerät selbst erlauben. Zwar ermöglichen nahezu alle Geräte die Ausführung von vorkompilierten Java-Midlets, die direkte Erzeugung und Ausführung von Programmquelltexten auf Mobiltelefonen ist jedoch bisher nur unter dem Betriebssystem Symbian S60³ möglich. Zur Ausführung von Python Quelltexten wird zum einen PythonForS60 mit der PythonScriptShell⁴ benötigt. Um nicht nur auf den interaktiven Modus der PythonShell beschränkt zu sein, wird noch ein beliebiger Texteditor benötigt. An dieser Stelle sei Ped [Wah07] empfohlen, der noch einige Zusatzfunktionen speziell zur Programmiersprache Python zur Verfügung stellt. Auch wenn zum Programmieren nicht direkt notwendig, so sollte als eines von vielen Zusatzprogrammen der Y-Browser [Sil07] genannt werden. Dieser ist ein Dateimanager, der nicht die Einschränkung der häufig mitgelieferten Programme aufweist, keinen Zugriff auf Systemdateien bzw. Systemverzeichnisse zu erlauben. Für diejenigen, die mit der Programmiersprache Python nicht vertraut sind, sei als eines von vielen Tutorials im Netz [Jak08] empfohlen, welches bereits die besonderen Module von PythonForS60 berücksichtigt. Für diejenigen, die – unabhängig vom verwendeten Informatiksystem – unsicher sind, ob die Programmiersprache Python Verwendung in der Schule finden könnte, sei [Led07] empfohlen, welcher im ersten Teil genau diesen Aspekt mit sehr positivem Ergebnis für Python beleuchtet.

³Zumindest nach den dem Autor bisher vorliegenden Daten.

⁴<http://sourceforge.net/projects/pys60>

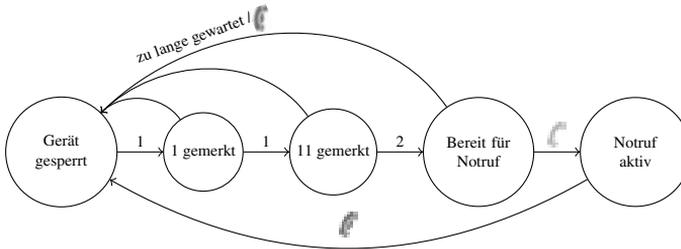
3.1 Konkrete Aufgabenvorschläge

Der Algorithmenbegriff. Die Bildungsstandards fordern in [GI08, S. 15] für Schülerinnen der Jahrgangsstufen 5-7, dass sie Handlungsvorschriften benennen, formulieren, lesen, verstehen und interpretieren können. Mit [GI08, S. 31, Bsp. 3.03] wird gleich ein passendes Beispiel zum Thema Mobiltelefone mitgeliefert. Eine weitere Formulierung mit anderem Fokus währe z. B.: *Notiere eine Handlungsvorschrift zum Verschicken einer SMS-Nachricht mit deinem Mobiltelefon, berücksichtige dabei, dass die Nachricht an ein oder mehrere Empfänger geschickt werden kann, und das die Empfängertelefonnummer sowohl manuell eingegeben, als auch aus der Kontaktliste gewählt werden kann.* Es bietet sich hier an, ein breites Spektrum verschiedener Mobiltelefone zu verwenden und danach Gemeinsamkeiten zwischen den Ergebnissen herauszuarbeiten. Hierbei kann in konstruktiver Weise mit unterschiedlichen Abstraktionsniveaus einer Pseudocode-Darstellung umgegangen werden (»Navigiere« oder »wähle« im Gegensatz zu »drücke linken Softkey« oder »drücke Taste *nach oben* auf dem Steuerungskreuz«), die notwendig wird, um Algorithmen betriebssystemübergreifend zu formulieren. Nach der Ausarbeitung der Ergebnisse kann insbesondere mit der Art und Weise verglichen werden, wie vorliegenden Bedienungsanleitungen bestimmte Schrittfolgen darstellen und erläutern.

Endliche Automaten. Während es bei Computern relativ schwierig ist, diesen als (nicht endlichen) Automat aufzufassen, da wenig konkrete Zustände zugeordnet werden können, fällt dieses bei Mobiltelefonen aufgrund ihrer Einfachheit wesentlich leichter. Ohne nähere Erläuterung werden aktivierte Profile wie *Lautlos* oder *Draußen* von Schülern und Schülerinnen als Zustände erkannt und auf diesen ersten intuitiven Formulierungen aufbauend wird die Einführung handfester Begriffe im Sinne der Automatentheorie weniger schwierig. Als mögliche Aufgabenstellung bietet sich die Beschreibung einer bestimmten Funktionalität an, die von sehr vielen Mobiltelefonen unterstützt wird: Trotz gesperrter Mobiltelefonatatur können mit dem Gerät Notrufe getätigt werden. Am Beispiel des Nokia 5500 wird in Abbildung 1 eine Möglichkeit der Darstellung in Form eines Zustandsübergangsgraphen gezeigt.

Diktiergeräte, MP3-Player und Einkaufslistenassistenten... Die meisten ersten Programmieraufgaben, welche den Fokus weniger auf das Modellieren, sondern auf den ersten Kontakt mit der Programmiersprache legen, haben nicht das Ziel, ein neuartiges Programm zum Lösen konkreter Fragestellungen zu erstellen, für das bisher keine anderen Produkte existieren. Dies ist zwar traurig und sollte nicht zur Gewohnheit werden, kann in bestimmten Fällen jedoch zu schnellen Erfolgserlebnissen führen, da die Aufgabenstellung übersichtlich und einfach ist. [ST07] bietet für viele kleine Aufgaben, die im Buch vorgestellt werden, passende Code-Schnipsel zum Herunterladen an. Diese Programme können von Schülerinnen und Schülern nachprogrammiert werden, je nach Schwierigkeitsgrad können bestimmte Anweisungen mit ihrer Bedeutung in verschiedenen Detailstufen vorgegeben werden, es kann jedoch auch auf die entsprechenden Abschnitte in der Dokumentation zur PyS60-API [Lau08] hingewiesen werden.

Geocaching. Eine spielerische Verarbeitung informatischer Inhalte kann mit GPS-Modulen – z. B. Mobiltelefonen mit integriertem oder externem GPS-Empfänger – realisiert wer-



1/2: Es wurde die Taste 1 bzw. 2 gedrückt

☎: Es wurde die Wähltaste gedrückt

📞: Es wurde die Auflegen/Abbruch-Taste gedrückt

Abbildung 1: Ein endlicher Automat – Notruf mit dem Mobiltelefon

den, die dazu genutzt werden, eine GPS-Schnitzeljagd durchzuführen⁵. Damit kann Informatikunterricht unter freiem Himmel realisiert werden. Analog zur Nutzung von Webquests, welche kein informatisches Konzept, sondern nur eine Unterrichtsmethode darstellen, steigt und fällt der Bezug zu den Bildungsstandards mit den gestellten Aufgaben. Im Sinne eines *Multi-Cache*, bei dem mehrere Aufgaben nacheinander oder parallel bearbeitet werden müssen, sorgt die erhöhte Mobilität der Mobiltelefone dafür, dass auch konkrete Implementierungsaufgaben bzw. Aufgaben, deren Lösung eine Programmierung erfordert⁶, gestellt werden können. Erweitert werden die Möglichkeiten, wenn mit eigenen Programmen auf die GPS-Positionsdaten zugegriffen werden kann. Dieses ist mit dem PyS60-Modul *positioning* einfach möglich.

(Bluetooth-)Chat: Das Verschicken von einzelnen Textnachrichten über Bluetoothverbindungen oder per SMS ist ohne Zusatzsoftware möglich. Die einzelnen Nachrichten verlieren jedoch ihren Zusammenhang und ein Gespräch, welches über mehrere SMS geführt wird, kann im Nachhinein nur schwer nachvollzogen werden. An dieser Stelle bieten sich verschiedene Aufgabenstellungen an. Zum Einen die Realisierung eines Bluetooth-Chat-Programm, welches eine Verbindung zu einem bestimmten Gerät aufbaut und die ausgetauschten Texte in chronologischer Reihenfolge anordnet, eine Funktion zum Abspeichern kann nach belieben eingebaut werden. Eine andere Möglichkeit wäre ein SMS-Chat-Programm, welches die auf dem Mobiltelefon gespeicherten SMS untersucht – dazu bietet sich das PyS60-Modul *inbox* an – und die zu einer Kommunikation mit einer bestimmten Personen gehörenden Nachrichten in chronologischer Reihenfolge auflistet. Je nach Realisierung kann dies durch eigene, der SMS angehängte Codes geschehen, eine Analyse der Nachrichtenattribute sollte jedoch für diesen Zweck ebenfalls genügen.

Informatik, Mensch und Gesellschaft. Unter dem Stichwort der *standortbezogenen Dienste* bekommt die eigene Position, der eigene Standort, eine zunehmend stärkere Bedeutung. Doch diese Position ist ein Datum, mit dem nicht leichtfertig umgegangen werden sollte.

⁵Näheres zu Geocaching z. B. unter <http://de.wikipedia.org/wiki/Geocaching>

⁶Als Beispiel: Welches ist die zweite Ziffer der 120ten Fibonaccizahl?

Die Frage *Wo sind meine Freunde?* mag noch relativ harmlos klingen, in dem Moment, in dem jedoch ein fremder Serviceanbieter genau weiß, welchen Weg eine bestimmte Person zurückgelegt hat, kann diese Information eine ganz andere Bedeutung bekommen. Gerade bei der Einführung des iPhone in Deutschland, welches die Positionsbestimmung anhand von GPS mit einer Internetflatrate verbindet und damit die Möglichkeit eröffnet, jederzeit sehr einfach Positionsdaten offen zu legen, ist es notwendig für dieses Thema zu sensibilisieren.

3.2 PyObjVG

In den Bildungsstandards sind Hinweise gegeben [GI08, S. 25], wie das objektorientierte Programmierparadigma bereits in frühen Jahrgangsstufen anschaulich mit Hilfe von Vektorgrafiken eingeführt werden kann. Mit der *Einfachen Objektorientierten Programmiersprache (EOS)* [Pab07] wurde eine eigene Programmiersprache mit einer leistungsfähigen grafischen Oberfläche verbunden, die speziell für den Unterrichtslichen Einsatz in der fünften bzw. sechsten Jahrgangsstufe konzipiert ist. Problematisch ist jedoch die fehlende Plattformunabhängigkeit, da EOS bisher nur für Windows-Systeme verfügbar ist. Während bei Grafikprogrammen wie Inkscape oder der Draw-Komponente von OpenOffice.org zumindest Versionen für Linux und Mac existieren, so haben dennoch alle Programme etwas gemeinsam: Die Plattformunabhängigkeit hört spätestens bei Mobiltelefonen auf! Ein positives Beispiel für Plattformunabhängigkeit stellt die Implementierung des Stifte- und Mäuse-Paketes für PyS60 dar. Es kam die Idee, diese Lernumgebung zu erweitern und – analog zum EOS-Konzept – Möglichkeiten zum Zeichnen und interaktiven Verändern von grafischen Primitiven einzubauen. Im Vergleich zu EOS steht neben der gewünschten Plattformunabhängigkeit aber auch eine einfache Möglichkeit der Erweiterung im Vordergrund, dass System darf nicht nur in der gegebenen Form anwendbar und damit auf die einführenden Klassenstufen beschränkt sein, sondern muss im Sinne des Spiralprinzips ebenfalls in höheren intellektuellen Niveaus seine Anwendung finden.

Betrachtet man, dass der Grundstein für eine flexible Erweiterbarkeit bereits mit der Auswahl der Programmiersprache Python gelegt wird, so bleibt als weiterer Punkt nur die Realisierung von Interaktivität. Dabei trennt sich PyObjVG in zwei Bereiche. Einmal die *Objektverwaltung*, welche – unabhängig von der Darstellung – für die Erstellung der grafischen Primitive verantwortlich ist, eine Gruppierung erlaubt und völlig plattformunabhängig ist. Zum anderen eine *Zeichenkomponente*, welche eine bestimmte Schnittstelle zur Verfügung stellt, mit deren Hilfe die grafischen Primitive sich selbst zeichnen können. Diese Schnittstelle ist jeweils auf die in der Python-Umgebung vorhandenen Zeichen-Module anzupassen – mitgeliefert wird jeweils eine passende Version, die Methoden der Stifte- und Mäuse-Bibliothek auf Desktop- bzw. PyS60-Systemen benutzt. Die Objektverwaltung ermöglicht nicht nur die Verbindung von Grafikobjekten untereinander zu Zwecken der Gruppierung, sondern ebenfalls die Verbindung mit einem Grafikzeichner-Objekt, wodurch die automatisierte Neuzeichnung und damit die geforderte Interaktivität ermöglicht wird. Für den Einsatz in einführenden Jahrgangsstufen kann das Grafikzeichnerobjekt im *Einfachen Modus* versteckt werden, so dass der Fokus auf die Erstellung konkreter Gra-

fikobjekte und Änderungen ihrer Attribute gesetzt werden kann, die Aktualisierung der Zeichnung erfolgt automatisch.

Als Bonus, der das Konzept der Trennung von Objektverwaltung und Zeichnung und die damit verbundenen Vorteile zeigen soll, ist eine spezielle SVGGrafikzeichner-Klasse implementiert worden. Diese abstrahiert die angesprochene Zeichenschnittstelle derart, dass anstelle eines Bildschirms SVG-konform in eine Datei gezeichnet wird. Zusammen mit eingeschränkten Import-Routinen der Objektverwaltung ist damit das Speichern und Laden der erzeugten Bilder im SVG-Format möglich.

4 Kritik und Fazit

Die im vorherigen Abschnitt angesprochenen Beispiele sollen Ansätze zeigen, die die am Anfang von Abschnitt 1 gestellte Frage wie folgt beantworten: Es gibt keine Notwendigkeiten, die den Einsatz von Computern unabdingbar machen.

Bei Unterrichtsphasen, in denen keine konkreten Programmieraufgaben gestellt werden, ist gerade die Nutzung einer großen Bandbreite unterschiedlicher Betriebssysteme interessant, welche durch die verschiedenen Mobiltelefontypen der Schülerinnen und Schüler dargestellt werden. In dem Moment, in dem jedoch Programmieraufgaben gestellt werden, wird diese Bandbreite stark eingeschränkt, nach bisherigen Kenntnissen des Autors ist es aktuell nur auf Mobiltelefonen mit Symbian S60 Betriebssystem – hauptsächlich vertrieben von der Firma Nokia – möglich, eine Python-Programmierungsumgebung zu installieren und ohne Zusatzgeräte zu nutzen. Eine solche monopolistische Abhängigkeit ist nicht gewollt. Aus der Sicht eines Befürworters der Mobiltelefone bleibt zu hoffen, dass mit der Verbreitung der Android-Plattform zumindest ein weiteres Betriebssystem dazu kommt, auf welchem die direkte Programmierung mit Python möglich ist.

Weiterhin wurde vernachlässigt, dass es im Normalfall keine vollwertigen Textverarbeitungen, Präsentationsprogramme oder Tabellenkalkulationen auf den mobilen Geräten gibt. Der Umgang mit den sogenannten Standardanwendungen, wie er auch in den Bildungsstandards gefordert wird, kann also nur sehr schwer mit Mobiltelefonen umgesetzt werden. Die Informatik kann keine direkten Nutzungskompetenzen im Bezug zum Computer vermitteln, darin eingeschlossen sämtliche nur dort ausführbare Anwendungssoftware. Dies ist im ersten Blick dramatisch, auf den zweiten Blick jedoch genau das, was von den Überlegungen dieser Arbeit intendiert wurde. Der Fokus soll weg vom Informatiksystem Computer geschoben werden, so dass den tieferen informatischen Inhalten eine größere Bedeutung zugesprochen werden kann.

Es ist zu beachten, dass die meisten Überlegungen, die in dieser Arbeit getätigt wurden, nicht sauber empirisch begründet werden können, sondern nur als theoretische Überlegung entstanden sind. Es ist daher sehr wichtig, neben dem Pilotprojekt in Bergkamen weitere Klassen mit Mobiltelefonen ins Leben zu rufen, die neues Analysematerial bieten, um – durchaus kritisch – weiterarbeiten zu können. Zur Zeit der Fertigstellung dieses Berichtes wird aktuell an der Universität Wuppertal eine Lehrerfortbildungsmaßnahme geplant, innerhalb welcher nicht nur ein oder mehrere Klassensätze programmierfähiger Telefone

zur Verfügung gestellt, sondern deren Einsatz mit entsprechenden Gesprächen und vorgefertigten Unterrichtsmaßnahmen universitär begleitet werden soll.

Literatur

- [BMF05] Gender-Datenreport – 1. Datenreport zur Gleichstellung von Frauen und Männern in der Bundesrepublik Deutschland, November 2005. Erstellt durch das Deutsche Jugendinstitut e.V. in Zusammenarbeit mit dem Statistischen Bundesamt unter der Leitung von Waltraud Cornelißen – BMFSFJ: Bundesministerium für Familie, Senioren, Frauen und Jugend – <http://www.bmfsfj.de/Publikationen/genderreport/root.html> – geprüft: 1. März 2009.
- [GI08] GI. Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik für die Sekundarstufe I, April 2008. Beschluss des GI-Präsidiums vom 24. Januar 2008 – veröffentlicht als Beilage zur LOG IN 28 (2008) Heft 150/151 http://www.gi-ev.de/fileadmin/gliederungen/fb-iaad/fa-ibs/Empfehlungen/bildungsstandards_2008.pdf – geprüft: 1. März 2009.
- [Jak08] Andreas Jakl. Symbian OS – Python/PyS60, May 2008. <http://www.symbianresources.com/tutorials/general/python/Python.pdf> – last visited 1st March 2009.
- [Lau08] Jukka Laurila. PyS60 Library Reference—Release 1.4.5 final. Manual—python symbian s60—nokia corporation, OpenSource.nokia.com, December 2008. http://downloads.sourceforge.net/pys60/PythonForS60_1_4_5_doc.pdf – last visited 1st March 2009.
- [Led07] Dominique Lederer. Python und Zope als Unterrichtswerkzeuge. Diplomarbeit – Master of Science in Engineering/Diplomingenieur (FH), Fachhochschule Technikum – Studiengang Informations- und Kommunikationssysteme, Wien, Mai 2007. http://www.stadtgespraeche.com/static/Python_und_Zope_als_Unterrichtswerkzeuge.pdf – geprüft: 1. März 2009.
- [MPF07] MPFS. JIM 2007. Jugend, Information, (Multi-)Media. Basisstudie zum Medienumgang 12- bis 19-Jähriger in Deutschland. Forschungsbericht, mpfs, Stuttgart, November 2007. MPFS – Medienpädagogischer Forschungsverbund Südwest <http://www.mpfs.de/fileadmin/JIM-pdf07/JIM-Studie2007.pdf> oder <http://www.handywissen.info/downloads/jimstudie2007.pdf> – geprüft: 1. März 2009.
- [MPF08] MPFS. JIM 2008. Jugend, Information, (Multi-)Media. Basisstudie zum Medienumgang 12- bis 19-Jähriger in Deutschland. Forschungsbericht, mpfs, Stuttgart, November 2008. MPFS – Medienpädagogischer Forschungsverbund Südwest http://www.mpfs.de/fileadmin/JIM-pdf08/JIM-Studie_2008.pdf – geprüft: 1. März 2009.
- [Pab07] Martin Pabst. EOS - Einfache objektorientierte Programmiersprache. online, Dezember 2007. <http://www.berg.heim.at/anden/420971/eos/index.htm> – geprüft: 1. März 2009.
- [Sil07] Jukka Silvennoinen. Y-Browser—filebrowser application designed for S60 3rd edition devices, July 2007. <http://www.drjukka.com/YBrowser.html> – last visited 1st March 2009.

- [ST07] Jürgen Scheible und Ville Tuulos. *Mobile Python. Rapid Prototyping of Applications on the Mobile Platform*. Wiley, Chichester, October 2007. <http://mobilepythonbook.com/> – last visited 1st March 2009.
- [Wah07] Arkadiusz Wahlig. Ped – Mobile Python IDE for Nokia S60 platform, 2007. Version 2.xx – <http://code.google.com/p/ped-s60/> – last visited 1st March 2009.

Konzepte, Modelle und Projekte im Informatikunterricht – Bewertungen und Erwartungen von Schülern und Studenten

Magnus Rabel, Reinhard Oldenburg

Institut für Didaktik der Mathematik und Informatik
Johann-Wolfgang-Goethe-Universität Frankfurt
Senckenberganlage 9
60325 Frankfurt
rabel@math.uni-frankfurt.de
oldenbur@math.uni-frankfurt.de

Abstract: Eine Befragung von Schülern in Informatikkursen und Studenten der Informatik wurde mit dem Ziel durchgeführt, Aspekte des Informatik-bezogenen Weltbildes der Testpersonen zu ermitteln. Dazu wurden Wertungen, Wünsche und Modellvorstellungen abgefragt. Die Ergebnisse geben Anregungen zur Weiterentwicklung des Informatikunterrichts.

1 Motivation der Untersuchung

Der Informatikunterricht hat trotz seiner jungen Geschichte schon viele grundlegende Veränderungen erlebt (Schubert/Schwill in [SS04]). Mit der objektorientierten Modellierung rückte in den letzten Jahren ein wichtiges, aber auch komplexes Thema in den Fokus des Interesses. Dabei stellt sich die Frage, ob der Informatikunterricht es schafft, dieses Thema so zu vermitteln, dass es nicht nur gelernt wird, sondern auch, dass die Schüler die Sinnhaftigkeit des Gelernten verstehen. Wird der Informatikunterricht in dem Sinne den Erwartungen der Schüler gerecht, dass sie Wissen erwerben, das ihnen nützlich erscheint und mit dem sie die Dinge tun können, die sie gerne tun möchten?

Wir gehen davon aus, dass das Erlernen komplexer Inhalte wesentlich mit der Konstruktion mentaler Modelle ([Du94], [Ed00], [Th02]) zusammenhängt. Mentale Modelle sind hypothetische Konstrukte der kognitiven Psychologie. Sie beschreiben innere Repräsentationen komplexer Systeme der Realität mit denen reale Vorgänge simuliert bzw. vorhergesagt werden können. Beim Lernen neuer Konzepte werden mentale Modelle erschaffen, die sich zumindest zu Anfang auf Analogien zu bekannten Konzepten stützen. Da mentale Modelle in ihrem Wesen immateriell sind, müssen sie auf irgendeine Weise nachgewiesen bzw. diagnostiziert werden. Die Diagnose mentaler Modelle ist mit Hilfe eines einzigen Fragebogens sehr schwierig und bedarf aufwändiger Diagnoseverfahren ([Th02], S. 33, [Du94], S.76, [GS83], S. 11). Daher beschränkten wir uns in einer ersten Annäherung auf die Abfrage konzeptueller Modelle, die die Lernenden im Laufe des Lernprozesses als hilfreich erfahren haben. Man kann davon ausgehen, dass unvorbereitet vor allem solche Modelle genannt werden, die sich als

dauerhaft nützlich erwiesen haben und deswegen häufiger Gegenstand des Denkens, also höchstwahrscheinlich zu mentalen Modellen der Schülerinnen und Schülern wurden.

Zur Beantwortung der eben gestellten Forschungsfragen haben wir einen Fragebogen entwickelt, der die Bewertung einiger Konzepte der Informatik durch Schüler und Studenten abfragt. Außerdem gaben die Probanden Auskunft über Modelle, die sie als nützlich in ihrem Lernprozess erlebt haben und zu Projekten, die sie gerne bearbeitet haben und solche, die sie gerne durchführen würden, wenn sie die nötigen Grundlagen dazu hätten. Es gibt einige andere Untersuchungen zu Einstellungen von Schülern (z.B. [SM05]) oder zur Vertrautheit im Umgang mit Computern (z.B. im Rahmen der internationalen PISA-Studien: [Pr01], [Pr04]), aber die hier fokussierten Fragen scheinen noch nicht untersucht worden zu sein.

2 Untersuchungsmethode

Diese empirische Untersuchung basiert auf einem Fragebogen, der zum Jahresende 2008 von 117 Schülerinnen und Schülern und 65 Studierenden bearbeitet wurde. Die Schülerinnen und Schüler entstammten zehn Informatikkursen der Jahrgangsstufen 12 und 13. Dabei handelte es sich sowohl um Grund- als auch Leistungskurse, deren Lehrerinnen und Lehrer sich dankenswerterweise bereit erklärten, die Fragebögen während der Unterrichtszeit ausfüllen zu lassen.

Die Schülergruppen stammen aus zwei Bundesländern und von neun verschiedenen Schulen. Die einzelnen Gruppen wurden in der Regel voll ausgeschöpft. Obwohl Repräsentativität in dieser Stichprobe nicht sichergestellt werden kann, gibt es keine Anzeichen, die gegen eine Annahme der Repräsentativität sprechen. Die studentische Vergleichsgruppe besteht aus 65 Hörerinnen und Hörern von Veranstaltungen für das dritte Fachsemester an zwei Universitäten, die von den Organisatoren der jeweiligen Übungsveranstaltungen die Gelegenheit bekamen, die Fragebögen zu bearbeiten. Die Wahl des dritten Semesters ist dadurch motiviert, dass diese Studenten bereits über ein Jahr in der Hochschulinformatik sozialisiert sind, und so ausreichend Zeit hatten, ihre Bewertungen aus Schulzeiten anzupassen. Andererseits liegt die Schulzeit dieser Studenten so kurz zurück, dass ihre schulischen Lernbedingungen wohl ähnlich waren wie die der aktuellen Schülerstichprobe. Man kann also davon ausgehen, dass Unterschiede zwischen den Gruppen wesentlich den weitergehenden Erfahrungen an der Universität zuzurechnen sind.

Der Fragebogen wurde nach dem Prinzip der intuitiven Konstruktion [MK07, S. 36] konstruiert. Der erste Teil des Fragebogens fragt die Lernjahre in Informatik und das Geschlecht ab. Im zweiten Teil sollten auf einer 6-stufigen Likert-Skala (0 bis 5) die folgenden Konzepte der Informatik bewertet werden. Aus Platzgründen geben wir schon in diesem Kapitel Ergebnisse an (Tabelle 1), nämlich die Mittelwerte jeweils für die Schüler, die Studierenden und die Gesamtpopulation. Grau unterlegte Zellen zeigen an, dass bei diesem Item der Unterschied zwischen Schülern und Studierenden auf dem 5%-Niveau statistisch signifikant ist (t-Test). Ein (+) in der jeweils zweiten Zeile innerhalb einer Zelle bedeutet einen Anstieg der Bewertung von Schülern hin zu Studenten.

Konzept	Ziel	Verständnis von Informatik bzw. Informatiksystemen (V) (SuS/Stud/Ges)	Erstellen von Software (E) (SuS/Stud/Ges)
Fallunterscheidung (bFallV / bFallE)		4,20 / 4,14 / 4,18	4,44 / 4,26 / 4,37
Schleife (bSchleifeV / bSchleifeE)		4,25 / 4,49 / 4,34 (+)	4,63 / 4,71 / 4,66 (+)
Prozedur bzw. Methode (bProzV / bProzE)		3,96 / 4,27 / 4,07 (+)	4,42 / 4,68 / 4,51 (+)
Rekursion (bRekV / bRekE)		3,61/3,92/3,73 (+)	3,66/3,77/3,70 (+)
Feld (=Array) (bArrayV / bArrayE)		3,62/3,90/3,72 (+)	4,04/4,14/4,08 (+)
Liste (bListeV / bListeE)		3,21/3,65/3,38 (+)	3,36/3,98/3,60 (+)
Stapel (bStapelV / bStapelE)		3,44/3,37/3,41	3,15/3,45/3,27 (+)
Graphische Darstellungen von Algorithmen (Struktogramme) (bStruktV / bStruktE)		3,47/3,44/3,46	2,66/2,62/2,64
Objekte mit Attributen und Methoden (bObjAttV / bObjAttE)		3,71/3,85/3,76 (+)	4,05/4,51/4,22 (+)
Objekte als Elemente einer Klasse (bObjKlaV / bObjKlaE)		3,55/3,49/3,53	3,88/4,32/4,05 (+)
Klassenbeziehungen wie Assoziation, Aggregation (bAssAggV / bAssAggE)		2,98/3,13/3,04 (+)	3,23/3,35/3,28 (+)
Vererbung (bVererbV / bVererbE)		3,36/3,60/3,45 (+)	3,56/4,02/3,73 (+)
Graphische Darstellungen von Klassenbeziehungen (UML) (bUmlV / bUmlE)		3,25/3,24/3,24	2,79/3,41/3,02 (+)
Automaten (bAutoV / bAutoE)		3,95/3,28/3,67	2,93/2,98/2,95 (+)
Graphische Darstellungen zur zustandsorientierten Modellierung (Zustandsdiagramme) (bDiaV / bDiaE)		3,61/3,37/3,51	2,68/2,95/2,79 (+)

Tabelle 1: Mittelwerte der Nützlichkeitsbewertungen von Schülerinnen und Schülern (SuS) bzw. Studierenden (Stud) und insgesamt (Ges), Itemkürzel in Klammern

Im zweiten Teil des Fragebogens wird abgefragt, welche Modelle Schüler und Studenten als wertvoll für Ihre Entwicklung kennen gelernt haben. Im Bewusstsein der eingangs genannten Schwierigkeiten haben wir uns hier für folgenden Prompt entschieden:

Schwierige Konzepte lernt man häufig an grundlegenden Beispielen oder mit Modellvorstellungen (z.B.: Variablen als Behälter für Werte). Geben Sie positive Beispiele aus Ihrem bisherigen Informatikunterricht (und –studium) an, als Sie selbst etwas Schwieriges auf diese oder ähnliche Weise gelernt haben:

Im dritten Teil des Fragebogens sollten die Schüler und Studierenden Projekte angeben, die aus ihrer Sicht schön und interessant sind:

- Geben Sie Beispiele von schönen oder interessanten Projekten bzw. Programmen an, die Sie dank Ihrer Informatikkenntnisse umsetzen konnten.
- Nennen Sie Informatikprojekte, die Sie gerne umsetzen würden, wozu Ihnen allerdings Ihrer Meinung nach noch die Mittel fehlen.

Die Antworten auf diese offenen Fragen wurden gesichtet und klassifiziert. An statistischen Analysen wurden hier nur Häufigkeiten bestimmt.

3 Ergebnisse und Interpretation

Im Folgenden werden die einzelnen Item-Gruppen des Fragebogens in Hinblick auf die oben formulierten Fragen analysiert.

Auswertung der Konzeptbewertungen

Bei der Testkonstruktion wurde entschieden, alle Konzepte doppelt bewerten zu lassen, nämlich in Hinblick auf das Verständnis von Informatiksystemen und auf die Konstruktion von Software. In der Tat zeigte sich, dass die Probanden diese Aspekte auch deutlich unterschieden haben. Beispielsweise waren bei den Schülern die Unterschiede zwischen den beiden Bewertungsformen der Konzepte bei fast allen Items signifikant bis auf Rekursion, Listen und Vererbung mit Effektstärken zwischen 0,35 und 0,53. Insgesamt war der Unterschied bei den Schülern mit einer mittleren Effektstärke von 0,24 ausgeprägter als bei den Studierenden. Dies kann man als Indiz deuten, dass mit zunehmender Erfahrung Verständnis und Konstruktion besser verzahnt werden. Mit anderen Worten: Im Laufe der Zeit stellt sich scheinbar die Erkenntnis ein, dass die erlernten Konzepte tatsächlich praxisrelevant seien.

Interessant ist zunächst die Frage, welche Konzepte in beiden Populationen am höchsten bewertet wurden. Unter dem Aspekt des Verstehens sind in beiden Populationen die am höchsten bewerteten Konzepte Fallunterscheidung, Schleife und Prozedur/Methode, also Konzepte der klassischen Algorithmetik.

Beim Erstellen von Software sind in beiden Teilpopulationen ebenfalls Schleifen und Prozeduren/Methoden unter den drei am höchsten bewerteten. Allerdings spielen unter

den Studierenden Objekte mit ihren Attributen und Methoden eine wichtigere Rolle. Dies bestätigt eine Vermutung, die wir schon vor Durchführung der Befragung hatten: Der Wert objektorientierter Konzepte zeigt sich vor allem bei größeren Projekten, zu denen es in der Schule selten (und an der Universität immerhin häufiger) kommt.

Die Ergebnisse zeigen, dass die Bedeutung „klassischer“ algorithmischer Aspekte durchweg hoch eingeschätzt wird. Diese hohe Wertschätzung der Algorithmik findet sich nicht nur bei Schülern und Studenten, sondern auch bei Forschern auf dem Gebiet der Informatik [ZS08]. Wir schließen daraus deskriptiv, dass es schon dem Unterricht in der Schule gelingt, die Bedeutung dieser Konzepte angemessen zu vermitteln, und normativ, dass die Schulung des algorithmischen Denkens nach wie vor ein zentrales Ziel des Informatikunterrichts sein sollte.

Eine Reihe von Unterschieden zwischen den Beurteilungen der Schüler und denen der Studierenden sind auf dem 5%-Niveau statistisch signifikant (t-Test)¹: Bzgl. des Verstehens sind das die Konzepte Prozedur, Liste und Automaten. Unter dem Aspekt „Erstellen von Software“ sind es Prozedur, Liste, Objekte sowohl als Elemente einer Klasse als auch mit ihren Attributen und Methoden und außerdem Vererbung und graphische Darstellungen / UML.

Bis auf eine Ausnahme werden alle genannten Konzepte unter dem jeweiligen Gesichtspunkt von den Studierenden als nützlicher eingestuft: Die Studierenden bewerten das Automatenkonzept als weniger nützlich für das Verständnis als die Schülerinnen und Schüler. Eine mögliche Begründung dafür liegt in der Auswahl der Probanden. An den ausgewählten Universitäten werden die Veranstaltungen zur theoretischen Informatik erst im vierten Semester angeboten, was zur Folge haben könnte, dass den ausgewählten Drittsemestern das Automatenkonzept nicht präsent ist.

Die Schülerpopulation ist keineswegs homogen. Insbesondere kennt jeder Informatiklehrer den Unterschied zwischen Autodidakten, die sich schon lange und intensiv mit dem Computer auseinandersetzen („Freaks“) und den eher „normalen“ Schülern, die wir hier Novizen nennen. Eine Unterscheidung ist anhand der Lernjahre möglich: Diejenigen, die in Informatik weniger als den Mittelwert von 3,5 Jahren angegeben haben, betrachten wir als Novizen, die die über diesem Wert liegen als Autodidakten. Auf dem 5%-Niveau ist nur der Unterschied bei der Bewertung „Graphische Darstellungen...Verständnis“ signifikant. Das liegt jedoch auch an den dann entstehenden kleinen Teilstichproben. Wenn man hier mit dem 10%-Niveau arbeitet, werden die folgenden Items signifikant unterschiedlich mit folgenden Effektstärken:

Item	Effektstärke
bListeV (p=0,09)	0.24
bStruktV" (p=0,08)	0.24
bAssAggV (p=0,09)	0.24
bUmlE (p=(0,06)	-0.26
bAutoV (p=0,06)	0.29

¹ Vgl. die grau unterlegten Zellen in Tabelle 1

In dieser Tabelle bedeuten positive Werte ein Ansteigen der Bewertung von Novizen zu Autodidakten. Bemerkenswert ist, dass dabei mit den Automaten ein Konzept der theoretischen Informatik deutlich gewinnt, während die graphische Darstellung von Klassenbeziehungen verliert. Hier schlägt sich offenbar die Erkenntnis nieder, dass man – mit entsprechender Erfahrung – auch ohne diese Methoden auskommt, während sie von Novizen als hilfreich erlebt werden.

In der Schülergruppe gab es keine geschlechtsbedingten Unterschiede. Bei den Studenten dagegen bewerten männliche Testpersonen den Beitrag von Feldern und Objektattributen signifikant höher.

Durch lineare Strukturgleichungsmodelle [Fo06] können Beziehungen zwischen Variablen erkundet werden. Dazu wurden die Summenvariablen $Algo = bFall + bSchleife + bProz + bRek$, $DS = bArray + bListe + bStapel$, $OO = bObjAtt + bObjKla + bAssAgg + bVererb$ jeweils für die Bewertung zum Verständnis und zur Entwicklung (gekennzeichnet durch den angehängten Großbuchstaben) getrennt berechnet. Durch hypothesengeleitetes Variieren ergab sich das in Abb. 1 gezeigte Strukturmodell, das die beste Modellpassung (adjusted Model fit nach R/sem: 0.75) liefert. Man erkennt, dass die Bewertungen von Datenstrukturkonzepten und von algorithmischen Konzepten stark zusammenhängen. Im Vergleich dazu ist die Bewertung der Konzepte der Objektorientierung nur schwach angekoppelt.

Eine Zwei²-Faktoren-Analyse der Gesamtpopulation zeigt eine Einteilung (siehe Abb. 2), die auch a priori nahe liegt: Es gibt einen Faktor, der vor allem aus algorithmischen Aspekten besteht (Bereich rechts unten im Diagramm), und es gibt einen anderen Faktor, der mehr strukturelle Vorstellungen beinhaltet (links oben). Die Variable lernj (Anzahl der Lernjahre) fällt – wie zu erwarten – aus diesem Schema heraus. Die gleiche Analyse angewendet auf die Teilpopulationen zeigt ein ähnliches Bild, allerdings sind dabei wegen der kleineren Fallzahlen die Faktoren nicht so deutlich getrennt.

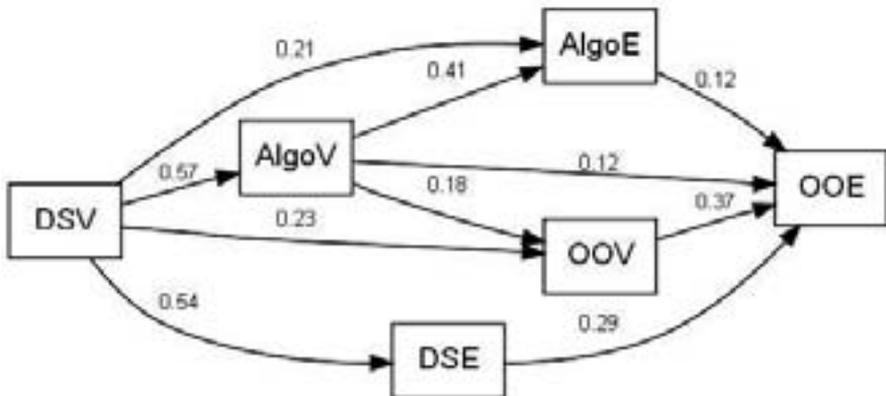


Abbildung 1: Strukturmodell

² Eine Analyse der Eigenwerte der Korrelationsmatrix mittels des R-Pakets nFactors zeigt, dass es gerechtfertigt ist, sich hier auf zwei Faktoren zu beschränken.

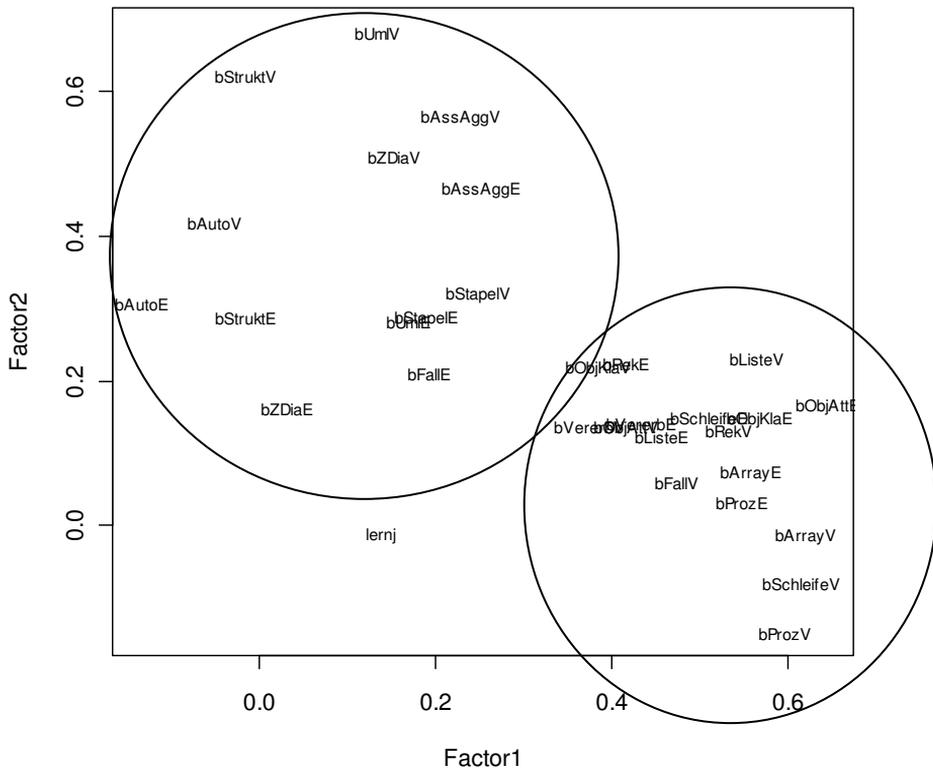


Abbildung 2: Faktorenanalyse

Auswertung der Modellnennungen

Die Frage nach den Modellen ist besonders interessant, weil die Probanden hier – allein dadurch, dass sie sich in der Befragungssituation an ein hilfreiches Modell erinnern – deutlich machen, dass dieses Modell wirklich einen Eindruck bei ihnen hinterlassen hat.

Wenn man die genannten Modelle in die Bereiche strukturierte Datentypen, Objekt-Orientierung und Algorithmen aufteilt, fällt deutlich auf, dass am häufigsten Modelle aus dem Bereich der strukturierten Datentypen genannt wurden (58 Nennungen). Das waren insbesondere analoge Alltagsmodelle zu Keller/Stack (31), Feld/Array (12), Queue (10) und Liste (5). Analogien zu Stack sind beispielsweise von Schülerseite ein „Stapel Papier auf dem Tisch“, ein „Tellerstapel“ oder in einer Formulierung, die auch die Funktionsweise expliziert, „Als Ablagestapel, der von oben wieder abgearbeitet wird“ und bei den Studierenden außerdem noch (als Einzelnenennung) die „Türme von Hanoi“. Für den Begriff des Feldes/Arrays wurden mehrheitlich eher abstrakte Analogien wie „Tabelle mit verschiedenen Werten“ oder „Koordinatenfeld“ aber auch seltener reale

Modelle wie ein „Schrank mit mehreren Schubladen“ oder ein „Schachfeld“ gewählt. Modelle aus dem Bereich der OO wurden am zweithäufigsten genannt (33). Das sind im Einzelnen vor allem Objekt-Klasse-Analogien aus der Realität (18) bzw. Alltagsbeispiele allgemein (7) und Analogien zur Vererbung (4). Als Beispiele mögen die folgenden Äußerungen eines Schülers und eines Studenten dienen: „Objekt-/Klassenkonzept: Klasse ist ein Bauplan, nach dem verschiedene Objekte gebaut werden können. So kann es z.B. die Klasse Auto geben mit den Eigenschaften Marke, Farbe, Alter, womit man viele verschiedene Objekte erzeugen kann, so z.B. einen 2 Jahre alten silbernen Volkswagen.“, „Schwierige Konzepte werden am besten durch konkrete Beispiele erlernt, z.B. bei der Vererbung von Klassen in Java an dem Beispiel einer Klasse Filialleiter, die von der Klasse Mitarbeiter erbt.“ Hier ergibt sich ein interessanter Unterschied: Im Bereich der strukturierten Datentypen sind die genannten Analogien bzw. Beispiele eher handlungsleitende Modelle als im Bereich der Objektorientierung. Beispielsweise sind die typischen Stack-Operationen durch die Analogie Ablagestapel nahezu vollständig erfasst, wohingegen die „Auto“-Analogie insbesondere für die Modellierung nicht-stofflicher Objekte (z.B. Objekt „Zugfahrt“) nicht tragfähig ist. Der Grund hierfür könnte darin liegen, dass der Objektbegriff zum einen grundlegender und allgemeiner ist als ein konkreter strukturierter Datentyp (z.B. Stack). Zum Anderen ist das Konzept für die Schülerinnen und Schüler noch relativ neu und entsprechend wenig verinnerlicht, sodass sie möglicherweise noch keine tragfähigeren und damit allgemeineren und abstrakteren Modelle nennen können. Auffällig war vor allem auch, dass die Schülerinnen und Schüler mehr Modelle nennen konnten bzw. nannten als die Studierenden. Durchschnittlich nannte ein Schüler 1,27 schwierige Konzepte, die er mit Hilfe von Modellvorstellungen oder Beispielen erlernt hatte, während es bei den Studierenden pro Kopf nur 0,72 Nennungen gab. Dennoch war die Reihenfolge der Bereiche bzgl. der Anzahl der Nennungen in beiden Populationen gleich.

Auswertung der durchgeführten hoch bewerteten Projekte

Insgesamt waren die häufigsten Nennungen:

	Projektart	Anzahl der Nennungen
1	Kleine Spiele (Tic Tac Toe, 4 gewinnt o.ä.)	49
2	Webseitenerstellung	33
3	Rechner (Taschen-, Währungs-, BMI,- etc.)	22
4	Datenbankanwendungen	18
5	Verwaltungsprogramme	9
	Verschlüsselung / Entschlüsselung	9

Auffällig ist auch in diesem Abschnitt in erster Linie, dass die Schülerinnen und Schüler mehr interessante Projekte nennen konnten als die Studierenden (1,97 zu 0,71 pro Fragebogen), wobei die Reihenfolge auch hier zumindest unter den häufigsten drei Nennungen identisch war. Allerdings wurden von den Studierenden mathematische Anwendungen wie Formelparser oder Funktionenplotter genauso häufig genannt wie Webprojekte.

Auswertung der Wünsche nach Projekten, die noch nicht durchführbar sind

Es war zu erwarten, dass die Nennungen zwischen den beiden Gruppen sehr weit auseinander liegen, denn die Studierenden haben im Laufe ihrer Ausbildung einerseits sicherlich schon einige Kompetenzen erworben, die die Schülerinnen und Schüler noch nicht aufweisen können. Andererseits sind ihnen durch das universitäre Umfeld auch größere Anwendungsfelder bekannt, als das in der Schule möglich wäre.

Insgesamt waren die häufigsten Nennungen:

	Projektart	Anzahl der Nennungen
1	Computerspiele	47
	davon 3D- oder Grafikspiele	15
2	Grafische Anwendungen	33
	ohne Spiele	18
3	Netzwerk-, WWW- oder Internetbasierte Anwendungen	20

Die Tendenz aus den vorherigen Abschnitten setzte sich auch in diesem letzten Abschnitt fort, sodass die Studierenden auch hier mit nur 0,48 Nennungen pro Kopf hinter den Schülerinnen und Schülern mit durchschnittlich 0,91 gewünschten Projekten zurückblieben. Darüber hinaus gaben die Studierenden im Vergleich zu den Schülerinnen und Schülern nicht den Spielen sondern den grafischen Anwendungen den Vorzug. Spiele wurden stattdessen genauso häufig genannt wie Webanwendungen. Die Wünsche der Schülerinnen und Schüler auf der einen Seite und die der Studierenden auf der anderen Seite liegen also nicht so weit auseinander wie ursprünglich angenommen, sondern konzentrieren sich in beiden Gruppen auf die genannten drei Bereiche.

4 Schlussfolgerungen

Auch wenn wie bei allen statistischen Erhebungen die Daten mit Vorsicht zu interpretieren sind, kann man doch einige Anregungen für den Unterricht wie für die Fachdidaktik gewinnen.

Für den Unterricht kann man zunächst festhalten, dass die Schüler eine ganze Reihe von Programmier-Projekten, die im Informatikunterricht weit verbreitet sind, als interessant und lohnend benannt haben. Das zeigt, dass diese Inhalte „ankommen“ und – ggf. mit Variationen – beibehalten werden sollten. Unter den offenen Wünschen sollten vor allem bedacht werden:

- Die Programmierung von Spielen wird von Schülern so deutlich nachgefragt, dass die Didaktik versuchen sollte, diesen Wunsch mit den im Unterricht umzusetzenden Fachinhalten kompatibel zu machen.
- 3D-Grafik – auch in Verbindung mit Spielen – sollte stärker thematisiert werden.
- Schüler wollen Netzwerke nicht nur verstehen, sondern webbasierte Projekte umsetzen.

Für die Fachdidaktik ergeben sich weitere Fragestellungen. Es wurden insgesamt viele mentale Modelle genannt, was bestätigt, dass sie für den Lernprozess wichtig sind. Die Fachdidaktik sollte ausarbeiten, wie diese im Unterricht noch pointierter entwickelt werden können. Insbesondere besteht offensichtlich ein Mangel an explizierbaren und tragfähigen mentalen Modellen zur Objektorientierung.

Aus der Bewertung der Konzepte kann man schlussfolgern, dass die Vermittlung der Objektorientierung noch nicht optimal ist. Erst Studierende bewerten – wohl vor dem Hintergrund der Erfahrung auch mit größeren Projekten – diese Konzepte höher.

Daraus und aus dem o.g. Mangel an mentalen Modellen in diesem Bereich kann man schließen, dass sich die Objektorientierung im Schulunterricht noch nicht voll entfaltet. Daraus kann man verschiedene Folgerungen ziehen, je nachdem welchen Stellenwert man der Objektorientierung im allgemeinbildenden Unterricht zubilligt. Eine mögliche Folgerung wäre, die Objektorientierung schon früher³ für die Modellierung möglichst breitgefächerter Problemstellungen einzusetzen und den Schülerinnen und Schülern im späteren Verlauf durch die Bearbeitung umfangreicherer Projekte die Gelegenheit zu geben, die Nützlichkeit der objektorientierten Modellierung intensiver zu erfahren.

Eine andere Herangehensweise könnte in einem genetischen Unterricht liegen, der Erfahrungen mit großen Strukturierungsproblemen mehr Zeit einräumt. Das hätte womöglich zur Folge, dass Problemlösen im Kleinen zunächst stärker betont wird, damit von den Schülerinnen und Schülern grundlegende algorithmische Fähigkeiten und Fertigkeiten erworben werden können, bevor sie mit Hilfe des objektorientierten Ansatzes umfangreichere und attraktive Probleme bearbeiten können. Insgesamt würde dadurch die gegenwärtig sehr starke Stellung der OOM zugunsten anderer Modellierungstechniken relativiert.

Eine wichtige Aufgabe ist es, diese Überlegungen mit den Wünschen der Schüler in Einklang zu bringen, so dass ein noch attraktiverer und zugleich fachlich solider Unterricht möglich wird, wobei auch die Erfahrungen von Informatiklehrkräften in die weiteren Untersuchungen eingebracht werden sollten.

Literaturverzeichnis

- [Di07] Diethelm, I.: "Strictly models and objects first" -- Unterrichtskonzept und -methodik für objektorientierte Modellierung im Informatikunterricht. Dissertation, Universität Kassel, Kassel, 2007.
- [Du94] Dutke, S.: Mentale Modelle: Konstrukte des Wissens und Verstehens. Kognitionspsychologische Grundlagen für die Software-Ergonomie. Verlag für Angewandte Psychologie, Göttingen, 1994.
- [Ed00] Edelman, W.: Lernpsychologie. 6. Auflage. Beltz PVU, Weinheim, 2000.
- [Fo06] Fox, J.: Structural Equation Modeling with the sem Package in R. Structural Equation Modeling, 13(3), 465–486.
- [GS83] Gentner, D.; Stevens, A.L. (Hrsg.): Mental models. [7. pr.]. Erlbaum, Hillsdale, NJ, USA, 1983.

³ Einen verhältnismäßig radikalen Ansatz verfolgt in diesem Zusammenhang Ira Diethelm, die in [Di07] unter anderem dafür plädiert, das objektorientierte Modellieren an den Anfang des Informatikunterrichts der Sekundarstufe II zu stellen und die Programmierung davon abzukoppeln.

- [MK07] Moosbrugger, H.; Kelava, A. (Hrsg.): Testtheorie und Fragebogenkonstruktion. Springer-Verlag, Heidelberg 2007.
- [Pr04] Prenzel, M. (Hrsg.): PISA 2003. Der Bildungsstand der Jugendlichen in Deutschland; Ergebnisse des zweiten internationalen Vergleichs. Waxmann-Verlag, Münster 2004.
- [Pr07] Prenzel, M. (Hrsg.): PISA 2006. Die Ergebnisse der dritten internationalen Vergleichsstudie. Waxmann-Verlag, Münster 2007.
- [SM05] Schulte, C.; Magenheim, J.: Novices expectations and prior knowledge of software development – results of a study with high school students. Proceedings ICER, Seattle, WA, USA, 2005, S. 143-153.
- [SS04] Schubert, S.; Schwill, A.: Didaktik der Informatik. Spektrum-Verlag, Heidelberg 2004.
- [Th02] Thomas, M.: Informatische Modellbildung. Modellieren von Modellen als zentrales Element der Informatik für den allgemeinbildenden Schulunterricht. Dissertation. Universität Potsdam, Potsdam 2002.
- [ZS08] Zendler, A.; Spannagel, C.: Empirical foundation of central concepts for computer science education. Journal of Educational Resources in Computing Vol. 8, No. 2, 2008, S. 1-15.

Wie gewinnt man Schülerinnen und Schüler für ein Informatikstudium? – Maßnahmen deutscher Hochschulen

Torsten Brinda, David van de Water

Didaktik der Informatik
Universität Erlangen-Nürnberg
Martensstr. 3
91058 Erlangen
{torsten.brinda|david.vandewater}@informatik.uni-erlangen.de

Abstract: Ausgehend von dem Wunsch und der Notwendigkeit, die Anfängerzahlen in den informatikbezogenen Studiengängen zu erhöhen sowie das Bild der Informatik ins rechte Licht zu rücken, wurden im Rahmen der vorliegenden Studie 143 deutsche Fachhochschulen und Universitäten nach ihren Maßnahmen befragt, mit denen sie versuchen, Schülerinnen und Schüler für ein Studium im Informatikbereich an ihrer jeweiligen Hochschule zu gewinnen. Dabei wurden die Befragten gebeten, bereits durchgeführte bzw. geplante Informations- und Werbeveranstaltungen, eingesetzte Informations- und Werbematerialien sowie die für deren Verteilung genutzten Kanäle mitzuteilen. Der vorliegende Beitrag enthält eine Auswertung und Klassifizierung der Erscheinungsformen basierend auf den Rückmeldungen von insgesamt 39 Hochschulen.

1 Motivation

Laut dem Statistischen Bundesamt sank die Zahl der Studienanfänger in der Informatik von 2000 bis 2006 um ca. 23,5% [SB08]. Seit 2007 ist wieder ein leichter Anstieg zu beobachten. Demgegenüber steht ein großer Fachkräftebedarf seitens der IKT-Industrie. Laut einer Studie des Bundesverbandes Informationswirtschaft, Telekommunikation und neue Medien e. V. (BITKOM) (Verbund von ca. 1.200 Unternehmen der IKT-Branche) [Bi07] waren Ende 2007 43.000 informatikbezogene Stellen in Deutschland unbesetzt, davon 18.000 in der IKT-Branche (IKT – Informations- und Kommunikationstechnologien) und 25.000 in Anwendungsfeldern. Auch die Politik hat diesen Fachkräftemangel auf der Agenda. So wurden Maßnahmen zur Beseitigung dieses Mangels in der Abschlusserklärung des 3. Nationalen IT-Gipfels der Bundesregierung (2008 in Darmstadt) [BW08] im Zielbereich „Begeisterung für IKT wecken“ mit den Unterpunkten „Nachwuchsförderung“ an Schulen, „Fachkräfteausbildung“ und Förderung von Frauen in „MINT-Berufen“ (MINT – Mathematik, Informatik, Naturwissenschaften, Technik) verankert. Im Spannungsfeld zwischen nicht zufriedenstellenden Studienanfängerzahlen im Bereich der Informatik, falschen Vorstellungen von Schülerinnen und Schülern über Informatik-Studieninhalte (vgl. [Ca06]) sowie dem negativ besetzten öffentlichen Bild von Informatikerinnen und Informatikern und dem von der Politik und der IT-Industrie geäußerten großen Bedarf an Fachkräften werden an vielen nationalen und internationalen Hochschulen Maßnahmen ergriffen, um mehr junge Menschen für ein Studium im

Informatikbereich zu gewinnen und diese bis zum erfolgreichen Abschluss zu halten (z. B. [HB01], [Mc02], [He04], [Bi07], [Wa08]). Mit dem Ziel durch eine deutschlandweite Dokumentation und Strukturierung von Studienwerbemaßnahmen den informatik- und technikbezogenen Fachbereichen der Fachhochschulen und Universitäten die Möglichkeit zu geben, ihre eigenen Maßnahmen zu reflektieren, ggf. zu optimieren und um damit die Zahl der Studienanfänger im Bereich der Informatik insgesamt positiv mit zu beeinflussen, wurden im November 2008 die 143 Vertrauensdozentinnen und -dozenten der Gesellschaft für Informatik e. V. (GI) nach den Maßnahmen ihrer jeweiligen Hochschulen zur Studienwerbung und -information im Bereich der Informatik befragt. Der vorliegende Beitrag gibt einen strukturierten Überblick über die Rückmeldungen von insgesamt 39 deutschen Hochschulen. Die Arbeit versteht sich damit auch als Beitrag und notwendige Voraussetzung für eine systematische und vergleichende Untersuchung der Wirksamkeit bekannter Maßnahmen, die in dieser Arbeit aber noch nicht erfolgte.

2 Forschungsmethodik

Im Wintersemester 2007/08 gab es laut dem Statistischen Bundesamt in Deutschland 104 Universitäten und 184 Fachhochschulen [SB08]. Die GI hatte Ende 2008 an 143 dieser Hochschulen jeweils einen Dozenten als Ansprechpartner (sog. GI-Vertrauensdozenten [BV09, 1]), was bei den 149 Informatik-Hochschulstandorten (Stand April 2008: [BV09, 2]) einer Abdeckung von 96% entspricht. Um einen aktuellen, nationalen Stand zu den Maßnahmen zur Studieninformation und -werbung im Bereich der Informatik zu erheben, wurden alle GI-Vertrauensdozenten im November 2008 über die GI-Geschäftsstelle per E-Mail um Beantwortung folgender Fragen gebeten:

1. *Welche Arten von Studieninformations- und -werbeveranstaltungen bietet die Informatik an Ihrer Hochschule an bzw. an welchen beteiligt sie sich?*
2. *Welche Arten von Informations- und Werbematerialien (spezielle Webseiten, Informationsbroschüren, Flyer, Plakate o. ä.) verwendet die Informatik an Ihrer Hochschule, um Schülerinnen und Schüler für ein Informatikstudium zu gewinnen?*
3. *Welche Informations-/Werbekanäle nutzt die Informatik an Ihrer Hochschule, um Schülerinnen und Schüler auf die jeweiligen Angebote aufmerksam zu machen?*

Bei der ersten Frage wurde ein Antwortschema vorgegeben. Dabei sollten für jede Veranstaltung *Bezeichnung, Kurzbeschreibung, Zielgruppe, Turnus* (z. B. jährlich, unregelmäßig), *Dauer* (wenige Std./eintägig/mehrtägig), *Anbieter* (Informatik/andere Einrichtung) sowie *URL(s)* angegeben werden. Ferner wurde bei allen Fragen um die Zusendung zusätzlicher PDF-Dokumente und/oder URLs gebeten, sofern verfügbar.

3 Ergebnisse

Die Ergebnisse basieren auf den E-Mail-Rückmeldungen von 39 Hochschulen (23 Universitäten, 16 Fachhochschulen), was einer Rücklaufquote von 27,3% entspricht. Sämtliche Hinweise zu konkreten Angeboten (z. B. Veranstaltungsformen, -titel) basieren auf

diesen Rückmeldungen sowie auf den Ende 2008/Anfang 2009 auf den referenzierten Webseiten (siehe [BV09]) zu findenden Angaben. Da hier Hochschulen befragt wurden, werden bei den Anbietern (soweit angegeben) nur diese bzw. deren Einrichtungen genannt, eventuelle weitere Mitanbieter (Verbände, Initiativen, Unternehmen etc.) sind den jeweiligen Webangeboten zu entnehmen. Für eine grafische Übersicht siehe [BV09].

3.1 Veranstaltungen zur Studienwerbung/-information in der Informatik

Da hier Hochschulen befragt wurden, findet die Mehrzahl der gemeldeten Angebote auch direkt dort statt. In der nachfolgenden Zusammenstellung werden Veranstaltungen in Kooperation mit und an *Schulen* sowie an *sonstigen Orten* entsprechend gekennzeichnet. An den *Hochschulen* lassen sich Veranstaltungsformate zur Studienwerbung/-information in der Informatik für Zielgruppen unterscheiden, deren Studienentscheidung entweder noch bevorsteht (Schülerinnen und/oder Schüler zwischen Grundschulalter und Studienbeginn) oder die andere bei der Studienentscheidung anleiten (Lehrkräfte, Verwandte, Freunde). Die im Rahmen der Studie identifizierten Erscheinungsformen werden nachfolgend nach diesen Zielgruppen differenziert dargestellt.

Veranstaltungen für Kinder (Grundschule, Beginn Sek. I): Für diese Zielgruppe organisieren einige Hochschulen sog. *Kinderunis*. Hierbei handelt es sich um kindgerechte Einzelveranstaltungen oder Veranstaltungsreihen, i. d. R. in der Verantwortung der jeweiligen Hochschule oder deren Studienservice- bzw. Presseabteilung, zu der Informatik-Abteilungen Programmpunkte beitragen. Beispiele für derartige Angebote sind Vorlesungen zu Themen wie „Mit dem Computer die Welt verstehen“, „Wie können Computer rechnen?“, „Wie funktioniert das Internet?“ oder die Aktion „Kann man etwas bauen, ohne dass es wirklich da ist?“ (vgl. [BV09, 7-17]). Das Ziel dieser Formate ist es, frühzeitig und spielerisch Informatikinteresse zu wecken und dauerhaft zu verankern.

Veranstaltungen für Schülerinnen und Schüler vor dem Abitur: Da für diese Zielgruppe die Studienfach- und -ortsentscheidung am greifbarsten ist, findet sich für diese auch das breiteste Veranstaltungsangebot. So gibt es zunächst vielfältige *Schnupperangebote* [BV09, 19-63]. Durch diese soll Schülerinnen und Schülern ein realistischer Einblick in die Wissenschaft Informatik, ihre Anwendungsbereiche und in die ihr zugeordneten Studienmöglichkeiten gegeben werden, damit diese eine spätere Studienfachentscheidung nicht nur aufgrund von persönlichem Technikinteresse, Anwendungserfahrungen oder externen Einflussgrößen (Informatikunterricht, Medien etc.) treffen. Die Benennung solcher Schnupperangebote ist vielfältig. Varianten heißen z. B. „Schnuppertage/-studium (Informatik)“, „Ferienkurs Informatik“, „SchülerCampus“, „Sommeruni/Summer University“, „Autumn School“, „(Schüler-/Studien-) Informationstage (Informatik)“, „Beratungstage“, „Hochschul-/Fakultäts-/Informatiktage“ oder „Tag der offenen Tür“. Die Hochschulen, Fakultäten oder Informatikeinrichtungen bieten solche Formate an der jeweiligen Hochschule als halb- oder eintägige Veranstaltung, als Kompaktveranstaltung an mehreren aufeinanderfolgenden Tagen oder auch als Folge von Einzelterminen an (z. B. Vortragsreihe, wie „Handy- und Pocket-PC-Programmierung für Einsteiger“). Die Informatikabteilungen beteiligen sich dabei mit fachspezifischen Beiträgen an Angeboten anderer Einrichtungen (z. B. Techniktag einer Technischen Fakultät, Studieninformationstag einer Hochschule) oder sind selbst Ausrichter. Die Grenzen

zwischen den Ausprägungen sind fließend und deren grundlegender Aufbau überall ähnlich. Angeboten wird eine Auswahl aus folgenden Bausteinen:

- *Einblick in Studieninhalte* durch Teilnahme an regulären Vorlesungen oder durch an speziell auf die Zielgruppe zugeschnittenen Überblicksvorträgen/Vorfürungen (z. B. „Mustererkennung in der Praxis: den Computer bedienen ohne ihn anzufassen – Gestensteuerung mittels einer Time-of-Flight-Kamera“),
- *Informationen zu den Studienangeboten inkl. Berufsaussichten* in Vorträgen und an Informationsständen, hierbei auch „Informationen aus Studierendensicht“,
- *Präsentationen von Projekten und/oder studentischen Arbeiten*,
- *Ausstellungen mit Präsentation von Hochschul-, Wissenschaftseinrichtungen und Unternehmen*,
- *Laborführungen*,
- *praktische/anwendungsorientierte, angeleitete Übungen/Experimente/Workshops* (z. B. zu „Computersehen: Analyse digitaler Bilder“, „Programmieren von Echtzeitanimationen“),
- *Gespräche mit Studierenden* (z. B. über „Das Informatikstudium – Studierende berichten“), mit *Wissenschaftlern, Praktikern, Berufstätigen* (z. B. „aus dem Berufsleben einer Informatikerin“), „Eignungsberatung in Gruppen“,
- *Aufbau von Firmenkontakten* (z. B. ein Vortrag zum Thema „Bewerbung um ein Praktikum, und zwar richtig!“),
- *(Bus-)Exkursionen zu Unternehmen, soziale Veranstaltungen* (wie Grillpartys).

Während die o. g. Schnupperangebote keine spezifischen Vorkenntnisse erfordern, gibt es auch solche, die bereits Informatikkenntnisse voraussetzen. So richtet sich das mehrtägige Probestudium Informatik der LMU München [BV09, 48] an Schülerinnen und Schüler mit „fundierte Programmierkenntnissen (grundlegende Kontrollstrukturen, Unterprogramme, Methoden bzw. Prozeduren oder Funktionen, Grundlagen in Objektorientierter Programmierung)“, die für die angebotenen Projektworkshops (z. B. „Voice over IP“, „Ähnlichkeitssuche in Bilddatenbanken“) und Vorlesungen (z. B. „Spiele in der Informatik“, „Suche im WWW“) benötigt werden. Andere Hochschulen verstehen unter Schnupperangeboten die *Öffnung regulärer Lehrveranstaltungen* für Studieninteressierte. Je nach Hochschule zeitlich offen oder begrenzt können Studieninteressierte so im Rahmen von Veranstaltungen wie „Systemprogrammierung“, „Datenbanken“ oder „Rechnerarchitektur“ eine Universität live erleben.

Neben den Schnupperangeboten gibt es vielfältige Angebote in Form von *Workshops und Projekten* [BV09, 64-71]. Dazu gehören schulische Betriebs-/Ferienpraktika an der Hochschule (z. B. zu Schwerpunkten wie „Einführung in die Struktur der Fakultät und den Rechnerbetrieb“, „Programmierung von Internetanwendungen“), hochschulseitig betreute Schüler-Projekte (z. B. Schüler-Projektlabor zu Themen aus dem Bereich des Schaltungsentwurfs wie „Elektronischer Schlüsselfinder“ oder „Elektronischer Würfel“), Schule in der Universität (z. B. Informatikunterricht an der Universität, wobei Schulen Teile ihres Unterrichts an die Hochschule auslagern) oder die Teilnahme von Schülern an der Präsentation studentischer Projektarbeiten.

Veranstaltungen für Schülerinnen: Da der Anteil weiblicher Informatikstudienanfänger im bundesweiten Schnitt zwar seit ca. 10 Jahren zwischen 15% und 20% liegt (vgl. [SB08]), an vielen Hochschulen aber auch deutlich darunter, werden viele Anstrengungen unternommen, um junge Frauen für die Informatik zu begeistern. Die Universität Siegen untersucht dazu in ihrem Projekt „Informatik ist weiblich“ [BV09, 77] in vielen

mit Schülerinnen in Schulen geführten Einzelinterviews, warum diese nur zu einem geringen Anteil Informatik studieren. Ziel ist es, auf dieser Basis erfolgsversprechende Maßnahmen zu konzipieren, durch welche die Zahl der Informatik-Studentinnen erhöht werden kann, auch vor dem Hintergrund, dass vielerorts bereits Aktivitäten stattfinden, die aber noch zu keiner substanziellen Veränderung beitragen konnten.

Viele Informatikabteilungen beteiligen sich mit Angeboten am bundesweiten Mädchen-Zukunftstag *Girls Day* [BV09, 79-95], dessen Ziel es ist, Schülerinnen den Blickwinkel auf eine Berufswahl zu erweitern, die über die „frauentypischen“ Berufe hinausgeht und die ihnen eine größere Chance zu beruflicher Entwicklung und Karrieremöglichkeiten im technischen und naturwissenschaftlichen Bereich bietet. Die Zielgruppe bilden allgemein Schülerinnen, wobei an einigen Hochschulen der Altersbereich eingeschränkt wird auf die Jahrgangsstufen 4/5-10, 9-13 oder ab 11. Ausgerichtet wird der Girls Day entweder von der jeweiligen Hochschule, den jeweils anbietenden Hochschul-Einrichtungen (z. B. der Informatik) und/oder der/den Frauenbeauftragten. Bei den Girls-Day-Angeboten lassen sich folgende typische Bausteine identifizieren:

- *Schnupperkurse/Workshops/Projekte* (z. B. zu Themen wie „Programmierung“, „Elektronische Kreide/interaktive Tafel“),
- *Führungen* (z. B. „Blick hinter die Uni-Kulissen“, Blick in Labore: „Wie fertigt man einen Mikrochip?“),
- *Vorträge/Infostände zu Studienangeboten in der Informatik*,
- *Gespräche mit Studentinnen, Mitarbeiterinnen und Professorinnen*,
- „*Take your daughters to work*“ als Möglichkeit, den Arbeitsplatz von Verwandten kennenzulernen.

Neben dem Girls Day bieten mehrere Hochschulen *Schnupperangebote und Informationsveranstaltungen im Bereich der Informatik für Mädchen* an [BV09, 96-106]. Neben monoedukativen Varianten der allgemeinen Schnupperformate (s. Tab. 1) werden darin beispielsweise auch spezifische Ringvorlesungen angeboten, bei denen Professorinnen und Professoren ihre jeweiligen Werdegänge und Forschungsschwerpunkte allgemein verständlich darstellen und Schülerinnen ermuntern, sich für diese Bereiche zu interessieren. Wesentlicher Unterschied gegenüber den allgemeinen Schnupperangeboten ist, dass die Betreuung überwiegend durch Frauen aus der Informatik mit Vorbildfunktion erfolgt, also Studentinnen, Mitarbeiterinnen und Professorinnen.

Des Weiteren werden an einigen Hochschulen *Workshops und Projekte für Mädchen* organisiert [BV09, 107-114]. Zu den Angeboten zählen Projekte und Projektstage (z. B. „Computer for Kids“ – „Wie räumt man seinen Computer auf?“ für junge Schülerinnen im Rahmen des Ada Lovelace Projekts [BV09, 107], Praktika/Workshops/Seminare/AGs für Mädchen (z. B. zu „Programmieren ohne Programmiersprache“, „Gesichtsanalyse“, „Geocaching“) und Veranstaltungen wie „LAN-Party Girls Only“ oder „I’m an IT-Girl“.

Die Hochschule Bremen bietet zudem für junge Frauen einen „*Internationalen Frauenstudiengang Informatik B. Sc.*“ an [BV09, 78].

Veranstaltungen für begabte/leistungsstarke Schüler: Mehrere Hochschulen sehen für diese Zielgruppe spezielle Angebote im Bereich der Informatik vor [BV09, 115-121], da man sich neben der Erfüllung des eigenen Bildungsauftrags durch deren Förderung auch eine frühzeitige Bindung später mutmaßlich besonders performanter Studierender an die jeweilige Hochschule erhofft. Die RWTH Aachen führt besonders ausgewählte Grund-

schüler (4. Klasse) unter dem Motto *Helle Köpfe in der Informatik* in Kooperation mit der Aachener Bürgerstiftung und regionalen Grundschulen in speziellen Veranstaltungen an Informatikthemen heran [BV09, 115]. Speziell an besonders begabte und leistungsstarke Schülerinnen und Schüler der Oberstufe (in Ausnahmefällen auch darunter) richten sich sogenannte *Frühstudienangebote* [BV09, 122-128], bei denen Schülerinnen und Schüler vor ihrem Abitur in geringem Umfang (z. B. 2-6 SWS) reguläre, einführende Universitätsveranstaltungen besuchen und die zugehörigen Prüfungen ablegen, deren Ergebnisse sie sich bei Aufnahme eines regulären Studiums auf Antrag anerkennen lassen können. Während bei diesen Angeboten reguläre Universitätslehrveranstaltungen für begabte Schüler geöffnet werden, bietet die FH Lausitz im Rahmen ihrer *Schülerakademie* [BV09, 128] an mehreren Einzelterminen „eine speziell erarbeitete Vorlesungsreihe für IT-interessierte und -begabte Schüler/innen [regionaler Gymnasien], welche wesentliche Inhalte der Informatik [...] widerspiegelt und weit über den Lehrstoff an den Gymnasien hinaus geht“. Themen in 2009 sind beispielsweise „Einführung in die digitale Bildverarbeitung“, „Sicherheit und Performance in Rechnernetzen“ oder „3D-Computergrafik“.

Eine individuellere Betreuung auf der Ebene von *Projekt- oder sonstigen Arbeiten* leistet z. B. die TU Chemnitz im Rahmen des Förderprojekts „BeLL – Besondere Lernleistungen“, was eine Betreuung besonderer Lernleistungen durch Hochschulmitarbeiter einschließt [BV09, 116].

Die Universität Potsdam (insbesondere das Hasso Plattner Institut – HPI) unterstützt regionale/nationale/internationale *Informatikschülerwettbewerbe*, z. B. die „International Olympiad in Informatics – IOI“ [BV09, 119], finanziell bzw. durch Bereitstellung von Räumlichkeiten und Gestaltung von Lehrgängen. Das HPI veranstaltete zudem einen „Tag der Talente“, bei dem 20 hochbegabte Schüler aus ganz Deutschland zu Gast waren. Die Preisträger unterschiedlicher Talentwettbewerbe des Bundes nahmen an einem Workshop teil, bei dem sie in die computergestützte Entwicklung von Hardware eingeführt wurden [BV09, 121].

Veranstaltungen für Jugendliche zwischen Abitur und Studienbeginn: Die FU Berlin [BV09, 129] bietet im Rahmen ihres Programms „Pro Informatik – Ein vorgezogenes Informatikstudium vor Beginn des ersten Semesters“ (Dauer 15 Wochen) Jugendlichen zwischen Abitur und Studienbeginn die Möglichkeit, Lehrveranstaltungen des ersten Studienjahres in Blockform zwischen Abitur und Beginn des ersten Semesters zu belegen und sich erbrachte Leistungen auf ein späteres Studium anrechnen zu lassen.

Veranstaltungen für Lehrkräfte: Einige Hochschulen gaben an, Lehrerfortbildungen im Informatikbereich anzubieten, z. B. [BV09, 130]. Gehofft wird einerseits, den Lehrerinnen und Lehrern dadurch ein zeitgemäßes und praxisnahes Bild der Informatik im Allgemeinen und an der jeweiligen Hochschule im Besonderen zu vermitteln, damit diese die Informationen als Botschafter an ihre jeweiligen Schüler tragen. Andererseits wird angestrebt durch die Bereitstellung von Unterrichtssequenzen den konkreten Unterricht indirekt fachlich und fachdidaktisch positiv mit zu beeinflussen.

Veranstaltungen für die Öffentlichkeit: Viele Hochschulen, Fakultäten und Informatikeinrichtungen richten zudem, teilweise in Kooperation mit Verbänden und Wirtschaftspartnern, *Veranstaltungen für die Öffentlichkeit* aus, wenngleich als Zielgruppe häufig besonders Studieninteressierte genannt werden [BV09, 131-149]. Zu den Erscheinungs-

formen zählen z. B. „Campusfeste“ oder „Tage der offenen Tür“ der jeweils anbietenden Einrichtung. Folgende typische Bausteine lassen sich bei diesen Formaten identifizieren:

- *Vorträge* (z. B. zu Themen wie „Bioinformatik & Systembiologie – Simulation der Zelle im Computer“, „Dr. med. Computer? Informatik in der Medizin“, „Kommunikationskontrolle im Netz“),
- *Vorfürhungen/Präsentationen* (z. B. zu Themen wie „First and Second Life – Fusion der realen und virtuellen Welt“),
- *Workshops/Experimente* (zu Themen wie „Was steckt im Computer/Netzwerk/Internet? Wir schrauben auf und schauen nach ...“),
- *Exponate, Informations- und Experimentierstände* (z. B. zur Studienberatung Informatik),
- *Führung durch die Informatik* (z. B. durch Labore),
- *Campusrundgang mit Studierenden* (Einblick in Werkstätten, Hörsäle u. ä.),
- *Bereichsbibliotheksbesichtigungen*,
- *Career Service* (z. B. mit Bewerbungshilfen für den Berufsstart),
- *Beratung über die informatikspezifischen Möglichkeiten eines Auslandsstudiums* (Akademisches Auslandsamt)
- *Informationsangebote der Fachschaft Informatik*,
- *„Treffpunkt Studentenleben“* (Studierende geben Auskunft und Tipps zu Studium und Studentenleben).

Neben besonderen Veranstaltungstagen finden auch Ringvorlesungen als *offene Vortragsreihen* zu aktuellen Themen der Informatik statt, die beispielsweise auch für Schülerinnen und Schüler und andere Interessierte geöffnet sind. Beispiele für Themen sind „Quanten-Computing“ oder „Systemarchitektur im Kraftfahrzeug“.

Einige Hochschulen beteiligen sich mit ihren Informatikeinrichtungen auch an regional organisierten *Langen Nächten der Wissenschaft* [BV09, 150-151] mit praxisorientierten Präsentationen und Mitmach-Experimenten für verschiedene Altersgruppen.

Veranstaltungen in Schulen: Bei Veranstaltungen in Schulen [BV09, 152-154] koordinieren Schulen und Hochschulen mitunter Informationstage, Vorträge (z. B. gehalten von Studierenden), Studienbasare (mit Informationsständen, Vorträgen) und Praktika oder ermöglichen Angebote wie „Rent-A-Prof“ (Professoren, die für einen Vortrag „gemietet“ werden).

Veranstaltungen an anderen Orten: In dieser Kategorie [BV09, 155-160] sind abschließend fachspezifische Exkursionen sowie Angebote auf Bildungsmessen (z. B. Abiturientenmessen) zu nennen.

3.2 Studienwerbe-/Informationsmaterialien im Bereich der Informatik

Bei den Studienwerbe-/Informationsmaterialien im Bereich der Informatik werden sowohl *elektronische* (Bereitstellung über Internet oder Datenträger) als auch *nichtelektronische Materialien* (insb. Printmaterialien) eingesetzt. Es kommen auch Mischformen zum Einsatz (beispielsweise Printmaterialien, die auch zum Download bereitstehen).

Internetangebote

An Schülerinnen und Schüler bzw. allgemein an *Studieninteressierte* richten sich Webportale und Webseiten, in denen eine Menge von Studienangeboten [BV09, 161-176] einer Einrichtung bzw. konkrete Studienangebote [BV09, 177-183] detailliert vorgestellt

werden. Diese Webangebote setzen sich typischerweise aus einer Auswahl folgender Bausteine zusammen, wobei beide Varianten nicht völlig überschneidungsfrei sind:

Überblicksportale	Studiengangswebseiten
<ul style="list-style-type: none"> - Überblick über den Studienort, - Überblick über das Studienangebot, - Beschreibung des Fachs/der Fächer, - Voraussetzungen für das Studium, - Einschreibe- bzw. Zulassungsverfahren, - Warum das Fach an diesem Ort studieren?, - Einblick in Forschungsprojekte, spätere Arbeitsfelder/Berufsaussichten, - Frauen und Informatik, - Berichte von Studierenden, - Studienfachberatung, - Informationen über Stipendien, - Frequently asked questions. 	<ul style="list-style-type: none"> - Angebote zur Studienorientierung, - Aufbau des Studiums/Verlaufsplan, - Ziele des Studiums, Voraussetzungen und Anforderungen, - Bewerbung und Zulassung, - Abschlussmöglichkeiten, - Regelstudienzeit, - Berufsfelder, - Ansprechpartner/Kontakt.

Tabelle 1: Typische Bestandteile von Webangeboten zur Studieninformation in der Informatik

Neben den Webseiten zur unmittelbaren Studieninformation bzw. verknüpft mit diesen gibt es ferner solche, die Studieninteressierte auf für sie geeignete *Veranstaltungsangebote zur Studieninformation* (vgl. 3.1) hinweisen [BV09, 184-185].

Andere Webseiten enthalten *nützliches Material/Entscheidungshilfen für Studieninteressierte*. Die RWTH Aachen ermöglicht Studieninteressierten beispielsweise, eine virtuelle Vorlesung zu besuchen [BV09, 186]. Zur Unterstützung der Studienfachwahl bieten einige Einrichtungen *webbasierte Eignungstests* in zwei Varianten an [BV09, 187-193]: mit der einen Variante testen Studieninteressierten allgemeine Fähigkeiten mit dem Ziel, eine Empfehlung für eine Studienrichtung (z. B. Technik) zu erhalten und mit der zweiten soll die Eignung für einen bestimmten Studiengang (z. B. Informatik) diagnostiziert werden. Einige Hochschulen setzen ferner *webbasierte Videos* zur Studienwerbung ein [BV09, 194-196]. So präsentiert sich die Universität Mainz beispielsweise in einem Infotrailer mit dem Titel „Warum Informatik in Mainz studieren?“. Der Betrachter erhält aus studentischer Perspektive einen Einblick in das Informatikstudium und in das studentische Universitätsleben. Ein Schwerpunkt liegt dabei auf der Korrektur des gängigen Klischees des Informatikers als unkommunikativen, introvertierten Fachsimpler.

Ein spezielles Angebot für *Lehrkräfte* findet sich auf den „Informatik – Material für die Schule (IMaS)“-Seiten der RWTH Aachen [BV09, 205]. Hier werden Informatiklehrpersonen Unterrichtsmaterialien zur Verfügung gestellt, welche von Lehramtsstudierenden während ihrer Studienzeit im Rahmen von Seminararbeiten, Staatsexamensarbeiten u. ä. entwickelt wurden.

Unter dem Motto „Universität für alle“ bietet die TU Berlin eine Webseite für *Interessierte jeden Alters* an [BV09, 206]. Hier finden sich u. a. Informationen zu „fachübergreifenden Studien“, „Lehrveranstaltungen für Hörer aller Fakultäten“, „Studium Generale“, „Vorlesungen/Ringvorlesungen“ oder „sonstigen Veranstaltungsreihen“.

TV-Aufzeichnungen

Von Seiten der Universität Kassel wurden zur Studienwerbung einige *TV-Aufzeichnungen*

gen gestaltet [BV09, 207-208], welche im hr-Fernsehen ausgestrahlt wurden und im Internet verfügbar sind. Dazu gehört sowohl ein Beitrag über Schülerstudenten an der Universität Kassel als auch ein Beitrag über einen Azubi-Unitag der Universität.

CD/DVD für Studieninteressierte/-anfänger

Weitere elektronische Werbe- und Informationsmaterialien werden auf *CD/DVD* verteilt. So bietet beispielsweise die Hochschule Heilbronn eine Erstsemester-CD an, auf der hilfreiche Informationen für den Einstieg in das Studium der Medizinischen Informatik enthalten sind. Diese CD wird einmal pro Semester Studienanfängern sowie interessierten Abiturienten ausgehändigt. Die Hochschule Mittweida stellt eine CD/DVD mit einem Studienführer inkl. Informationen zu allen dort angebotenen Studiengängen bereit.

Flyer und Broschüren: Bei den nicht-elektronischen Materialien sind Flyer [BV09, 209-218] und Broschüren [BV09, 219-227] ein verbreitetes Werbematerial. Teilweise werden diese auch digital zum Download bereitgestellt. Inhaltlich werden in diesen folgende Bereiche abgedeckt: Überblick über eine Menge von Studienangeboten, detaillierte Beschreibung konkreter Studienangebote, Überblick über Angebote für Schüler und Lehrkräfte, Beschreibung einer Einrichtung.

Die größte Gruppe der *Flyer und Broschüren* gibt einen Überblick über eine Menge von Studienangeboten oder beschreibt ein konkretes Studienangebot detailliert. Diese werden auf verschiedenen Ebenen angeboten, z. B. landesweit (Studienführer „Studieren in Hessen“ [BV09, 221]), hochschulweit (Broschüre der zentralen Studienberatung der Universität Mainz mit allgemeinen Informationen und Hilfestellungen zur Studienwahl [BV09, 219] und Antworten auf Fragen, wie „Warum studieren?“, „Was studieren?“ oder „Wie würde ein Studium in diesen Fächern aussehen?“), fakultätsweit oder einrichtungsweltweit (Studium Informatik an der Fernuniversität Hagen [BV09, 222] mit Hinweisen zu den Punkten „Allgemeines zum Studium“ (wie Studienmaterialien, Kursbelegungen), „Die Studiengänge“ (Darstellung der angebotenen Studien- und Abschlussmöglichkeiten) sowie „Informationsmöglichkeiten“). Typische Bestandteile von Studieninformationsflyern in der Informatik sind:

- | | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> - Hochschulortsspezifisches/Vorteile eines Studienortes, - (alternative) Studiermöglichkeiten, - Darstellung des Fachgebietes, - Studienaufbau/Studienverlaufsplan, - Studieninhalte, - Studien-/Prüfungsordnung, - Voraussetzungen und Zulassungsbeschränkungen/NC, - erforderliche Sprachkenntnisse, - Unterrichtssprache, | <ul style="list-style-type: none"> - Bewerbungsverfahren, - Studienbeginn, Studiendauer, - Studiengebühren, - Abschlussmöglichkeiten, - Auslandssemester, - Berufsbild, - Praxisrelevanz/Zukunftsperspektive/Berufsaussichten, - Kontaktadressen, - Impressum und Linkliste. |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Flyer und Broschüren zum *Veranstaltungsangebot für Schüler und Lehrkräfte* enthalten kalenderartige Aufstellungen von Veranstaltungen zur Studieninformation im Bereich der Hochschule, der Fakultät oder der Informatikeinrichtung, z. B. über Schülerinformationstage, Girls Days, Projektstage. Seitens der RWTH Aachen wird beispielsweise die Broschüre „Wissenschaft macht Hochschule“ mit einer zentralen Zusammenfassung aller Informationsangebote für Schüler angeboten [BV09, 223]. Darin enthalten sind

mitunter Hinweise auf „Informationstage“, „Angebote für Grundschüler/innen“, „Angebote für Mittel- und Oberstufenschüler/innen“, „Schülerpraktika“ sowie „Angebote für Lehrerinnen und Lehrer“. Ergänzt wird dieses Angebot durch eine Broschüre mit einem Schülervorlesungsverzeichnis.

Das HPI meldete zudem eine Informationsbroschüre mit einer *Beschreibung der eigenen Einrichtung*.

Poster: Auch Poster werden als Werbematerialien eingesetzt [BV09, 228-230]. Einerseits wird damit auf Studieninformations/-werbeveranstaltungen aufmerksam gemacht, andererseits auf die fachspezifischen Studienangebote. Die Universität Erlangen-Nürnberg verknüpft hierbei prominente Anwendungsgebiete der Informatik („Informatik und Medizin“, „Informatik und Sport“, „Informatik und Schule“, „Informatik und Unterhaltung“ sowie „Informatik und Automobiltechnik“) mit ihren eigenen Studienangeboten.

Ausstellungen: Seitens der TU Darmstadt wird eine Wanderausstellung „Abenteuer Informatik – Informatik begreifen“ angeboten [BV09, 231]. Diese ist für Besucher jeden Alters konzipiert. In der Ausstellung können die Besucher beispielsweise 15 Informatikexperimente ausprobieren.

3.3 Werbe-/Informationskanäle

Im Rahmen *persönlicher Kanäle* werden direkte Kontakte zu Mitgliedern der Zielgruppe gesucht. Diese entstehen hauptsächlich durch *persönliche Auskünfte* seitens der Studienberatung der Bundesagentur für Arbeit oder durch Studiengangkoordinatoren, *Veranstaltungen* wie Messen, Schülerakademien, Studentenclubs, Vorträge, Demonstrationen, Fortbildungen für Informatiklehrpersonen sowie *Kooperationen* zwischen Schulen und Hochschulen in der Lehre. *Nicht-persönliche Kanäle* bilden Kontakte über das *Internet* durch Webseiten, Mailinglisten, Newsgruppen, Anzeigen, Netzwerktreffen, über die *Presse* durch Pressestellen, Presseverteiler, Zeitungen, Radiointerviews, Anzeigen, regionale TV-Aufzeichnungen, über die *Post* durch Briefe an Schulen, über *Medien* wie Flyer, Poster (z. B. an schwarzen Brettern, in Schaukästen, in S-/U-Bahnen) sowie über *Institutionen* durch Kontakte zur Bildungsagentur.

4 Verwandte Arbeiten

Dass weitere Bemühungen um Studienanfänger im Fach Informatik und um die Vermittlung eines korrekten Informatikbildes erforderlich sind, zeigt beispielsweise eine Studie von Carter [Ca06]. In dieser wurden 836 US-amerikanische Schülerinnen und Schüler befragt, ob sie ein Informatikstudium aufzunehmen beabsichtigen sowie nach ihren Vorstellungen darüber, was Informatikstudierende lernen. Ein zentrales Ergebnis der Studie war, dass Schülerinnen und Schüler deswegen kein Informatikstudium aufnehmen, weil sie entweder ein inkorrektes Bild oder erst gar keine Vorstellung über ein solches haben. In der Vorstellung von Studienanfängern reduziert sich die Informatik oft auf die Nutzung und Administration von Computern, Hardware und Programmierung [MW06]. Dass auch informatische Bildung in Schulen dieses Problem nicht per se löst, zeigte eine

Untersuchung von Greening [Gr98]. Befragt wurden Schülerinnen und Schüler im Informatikunterricht mit folgendem Ergebnis: 92,9% besaßen zwar einen Computer und nutzten diesen auch zuhause, aber mehr als 58% waren nicht in der Lage, wenigstens näherungsweise anzugeben, was Informatik eigentlich ist. Viele der von den Hochschulen ergriffenen Maßnahmen (auch die im Rahmen der vorliegenden Arbeit gesammelten) haben deshalb das Ziel, einen möglichst breiten und unverfälschten Einblick in Teilbereiche der Informatik zu geben.

Aus der Vielfalt der Maßnahmen, über die in der Literatur berichtet wird, werden nun abschließend noch einige skizziert, die auch zeigen, dass an geeigneten Strategien bereits seit Jahren gearbeitet wird. Im Kontext von *Werbeveranstaltungen* für Schüler betont Walker [Wa08] den Wert von Schnupperangeboten und von Schulbesuchen (vgl. auch [Br88]). Schnupperangebote werden auch international z. B. als Sommeruniversitäten mit einer Mischung aus Einführungen, Workshops, Praktika und Freizeitgestaltung organisiert. Wo dies möglich ist, können die Teilnehmer in Studentenwohnheimen auf dem Campus übernachten und sich in der Mensa verpflegen. Ebenso gibt es internationale Erfahrungen mit Camps für junge Mädchen (im Alter von 10 bis 14 Jahren), wie z. B. eine sog. „Technologie Akademie für Mädchen“ [Gr08], deren Ziel es ist, Mädchen für den MINT-Bereich zu begeistern. In allen Praktika wurde ein altersgemäßes, praktisch angelegtes Projekt durchgeführt, um verschiedene mit dem Computer verbundene Berufe zu erkunden. Die Betreuung der Schülerinnen erfolgte überwiegend durch Frauen, was auch in [HB01] befürwortet wird. Bei den *Werbematerialien* wird der Wert von speziellen Webseiten und Fachbereichsflyern unterstrichen [Wa08], [Mc02] empfiehlt allerdings, alle Werbematerialien auf das darin kommunizierte Bild der Informatik hin zu überprüfen. Bei den *Werbekanal*en schließlich wird der persönliche Kontakt betont. McGrath Cohoon schlägt eine Vernetzung mit Lehrkräften, den Aufbau von Netzwerken und die Identifizierung und Positionierung von Vorbildern vor, die als Verteiler in Schulen und auf Werbeveranstaltungen eingesetzt werden können [Mc02]. Walker regt das Sammeln von Adressen Studieninteressierter an, um diese direkt zu kontaktieren [Wa08].

5 Zusammenfassung und Ausblick

Die Analyse der Ergebnisse der Umfrage zeigt eine breite Palette an Aktionen, die bundesweit durchgeführt werden, um die Anzahl der Einschreibungen sowie Abschlüsse in der Informatik positiv zu beeinflussen. Nach Abschluss der Auswertung werden die detaillierten Ergebnisse an diejenigen Hochschulen weitergeleitet, die einen Informatikstudiengang anbieten, so dass diese die Möglichkeit erhalten, mit Hilfe der hier dargestellten Vielfalt an Aktionen deren eigene anzupassen und ggf. zu optimieren. Erforderlich ist weiterhin, die Effektivität der Praktiken in Relation zu deren Kosten zu analysieren, da einige zwar beinahe kostenneutral sind (z. B. das Öffnen regulärer Lehrveranstaltungen für Schüler), andere jedoch signifikante Kosten verursachen (z. B. die Dopplung ausgewählter Lehrveranstaltungen). Über die Wirksamkeit solcher Maßnahmen ist bislang nur wenig bekannt. Einzelne Hochschulen (z. B. die Univ. Erlangen-Nürnberg) befragen ihre Anfänger nach den tatsächlichen Einflussgrößen ihrer Studienentscheidung. Hier werden z. B. vorrangig Eltern und Freunde benannt. Alle sind massiv beeinflusst durch die Medien, die oft weniger über die Erfolge der Informatik für die Men-

schen, als über aktuelle Gefahren durch Informatik (z. B. Kommunikationskontrolle im Netz) berichten. Informatikaktive sind in der medialen Berichterstattung zudem gerne weniger Identifikationsfiguren für die zukünftige Generation, als vielmehr sozial problematische aufgestellte Persönlichkeiten – Medienformate, wie „Das Model und der Freak (Pro7)“ lassen grüßen. Eine grundlegende Änderung dieser Wahrnehmung wird nur durch eine frühzeitige, hochqualitative informatische Bildung in Schulen durch gut ausgebildete Lehrer und eine intensive Zusammenarbeit mit den Medien möglich sein.

Literaturverzeichnis

- [Bi07] Bundesverband Informationswirtschaft, Telekommunikation und neue Medien (BITKOM e. V.): Standortnachteil Fachkräftemangel: Fakten und Lösungsansätze, 2007. URL: http://www.bitkom.org/files/documents/BITKOM_Positionspapier_Fachkraeftemangel_IT-Gipfel_2007.pdf (gepr. 16.02.2009)
- [Br88] McBride, W. E.; Calhoun, J.; Richards, J. L.; Taylor, H. G.; Garnet Walters, F: Recruiting more computer science students – what to do after the “glamour” has gone away? In: ACM SIGCSE Bulletin 20, 1 (February 1988), 1988; p. 181.
- [BV09] Brinda, T.; Van de Water, D.: Wie gewinnt man Schülerinnen und Schüler für ein Informatikstudium? - Ein Überblick über Maßnahmen deutscher Hochschulen: Internetquellen, 2009. URL: <http://ddi.informatik.uni-erlangen.de/Forschung/Projekte/StudWerb/> (gepr. 16.02.2009)
- [BW08] Bundesministerium für Wirtschaft und Technologie (Hrsg.): Dritter Nationaler IT-Gipfel: In Deutschland die digitale Zukunft gestalten. Darmstädter Erklärung vom 20. November 2008. URL: <http://bmwi.de/BMWi/Navigation/Service/publikationen,did=279984.html> (gepr. 16.02.2009)
- [Ca06] Carter, L: Why students with an apparent aptitude for computer science don't choose to major in computer science. In: Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education, ACM Press, New York, 2006; pp. 27-31.
- [Gr98] Greening, T. 1998. Computer science: through the eyes of potential students. In: Proceedings of the 3rd Australasian Conference on Computer Science Education. ACM Press, New York, pp. 145-154.
- [Gr08] Groth, D. P.; Hu, H. H.; Lauer, B.; Lee, H.: Improving computer science diversity through summer camps. In: Proceedings of the 39th SIGCSE technical symposium on Computer science education, ACM Press, New York, 2008; pp. 180-181.
- [HB01] Hassoun, S.; Bana, S.: Practices for Recruiting and Retaining Graduate Women Students in Computer Science and Engineering. In: Proceedings of the International IEEE Conference on Microelectronic Systems Education, 2001; pp. 106-107.
- [He04] Henry, T. R.; Holz, H.; Steinback, C.; Reed, C.; Baid, A.: Work in Progress – Student Retention and Recruitment in Computer Science Programs. In: Proceedings of the ASEE/IEEE Frontiers in Education Conference, 2004.
- [Mc02] McGrath Cohoon, J.: Recruiting and retaining women in undergraduate computing majors. In: ACM SIGCSE Bulletin 34, 2 (June 2002), 2002; pp. 48-52.
- [MW06] Maaß, S.; Wiesner, H.: Programmieren, Mathe und ein bisschen Hardware...Wen lockt dies Bild der Informatik? In: Informatik Spektrum 29 (2006) 1, S. 125-132.
- [SB08] Statistisches Bundesamt (Hrsg.): Bildung und Kultur. Schnellmeldungsergebnisse der Hochschulstatistik zu Studierenden und Studienanfänger/-innen. Vorläufige Ergebnisse, Wintersemester 2008/09. URL: <http://www.destatis.de/> (gepr. 16.02.2009).
- [Wa08] Walker, H. M.: Advertising and recruiting. In: ACM SIGCSE Bulletin 40, 2 (June 2008), 2008; pp. 16-17.

Prozessbegleitende, automatisierte Identifizierung der Problemlösestrategien von Lernenden beim Lösen algorithmischer Probleme

Ulrich Kiesmüller

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

Abstract: Beim Lehren algorithmischer Grundbegriffe werden oft spezielle Lern- und Programmierumgebungen eingesetzt. Diese besitzen altersgerechtes Design und sind leicht erlern- und bedienbar. Trotz allem haben viele Lernende Schwierigkeiten in diesem Bereich und ihre Leistungen bleiben hinter den Erwartungen zurück. Eine Voraussetzung zur Verbesserung des Lehr-/Lern-Prozesses besteht darin, genauere Kenntnisse bezüglich der von den Lernenden eingesetzten individuellen Problemlösestrategien zu erhalten. Ein Ziel dieses Forschungsprojekts ist es, bereits während des laufenden Lösungsprozesses die einzelnen Problemlösestrategien automatisiert zu identifizieren und zu kategorisieren. Dazu müssen in die Lernumgebungen spezielle Analyse- und Diagnosemodule integriert werden, deren Ergebnisse schließlich in der Gestalt von individualisierten und an die Vorgehensweise der Lernenden adaptierten Systemrückmeldungen verwendet werden können. Dies kann zu einer Verbesserung der Lern- und Programmierumgebungen und damit der Informatikstunden, in denen sie eingesetzt werden, führen. In diesem Artikel werden Untersuchungswerkzeuge und der Untersuchungsprozess erläutert sowie erste Studien beschrieben, deren Ergebnisse vorgestellt und diskutiert werden. Außerdem werden Möglichkeiten zur Verifizierung und Validierung dieser Ergebnisse aufgezeigt und eine davon ausgearbeitet.

1 Motivation

Das Erlernen der grundlegenden Ideen der Algorithmik und auch das Verständnis für das Programmieren rufen bei den Lernenden immer wieder Probleme hervor. Indizien hierfür sind hohe Durchfallquoten in Algorithmik-Anfängervorlesungen an den Universitäten sowie schlechte Schulnoten im Fach Informatik in diesem Bereich. Um die Schwierigkeiten so gering wie möglich zu gestalten, wird Programmieren oft unterrichtet ohne zu verlangen, dass die Lernenden Programmcode schreiben müssen. Die Vermittlung der grundlegenden Prinzipien der Algorithmik ist bereits im Bereich der Sekundarstufe ein fundamentaler Aspekt [Sc97]. Hierbei werden den Lernenden spezielle didaktisch reduzierte text-basierte (z. B. Karol, der Roboter [Pa94]) oder visuelle Programmiersprachen (z. B. Scratch [Ma04] oder Kara, der programmierbare Marienkäfer [Re03]) zur Verfügung ge-

stellt, um ihre ersten Schritte auf dem Gebiet der Programmierung zu unternehmen. In einigen deutschen Bundesländern werden die Grundlagen der Algorithmik bereits in der 7. Jahrgangsstufe gelehrt, wobei oben erwähnte altersgerecht gestalteten Lern- und Programmierumgebungen eingesetzt werden, durch deren Design sich Lernende motivieren lassen. Sie sind auf Grund der einfachen Erlern- und Bedienbarkeit innerhalb nur weniger Unterrichtsstunden bereits in der Lage, auch komplexe Aufgabenstellungen zu bewältigen. Zu deren Lösung existiert kein „optimaler, einzig richtiger“ Weg. Während die Lernenden ihre selbst erstellten Programme durch Ausführung testen, kommt es meist zu Fehlermeldungen des Systems. Diese rein technischen Rückmeldungen werfen bei den Lernenden während des Problemlöseprozesses neue Fragen auf, weshalb sie auf Hilfe von außen - in diesem Falle der Lehrkraft - angewiesen sind. Falls diese nun nur technische Hilfe gibt und den jeweils letzten Lösungsschritt der Lernenden korrigiert, um sich möglichst schnell an die korrekte Lösung anzunähern, bietet die Lehrkraft keine echte Alternative zu den Systemrückmeldungen. Ebenso stellt es kein Ziel der Didaktik der Informatik dar, die Lernenden lediglich dazu zu bringen, die Musterlösung schrittweise abzuschreiben.

Diese Feedbackvarianten führen seitens der Lernenden zu Unselbstständigkeit, weiterer Hilfebedürftigkeit und schließlich dazu, dass sie eine anfangs eingeschlagene Problemlösestrategie ziellos wechseln. Alle Faktoren bergen auch die Gefahr der Frustration der Lernenden. Um diese zu vermeiden, sollten sie auf ihre individuelle Vorgehensweise fokussiert und bei ihrer Beibehaltung unterstützt werden. Aus konstruktivistischer Sicht sollte die Lehrkraft bereits vorhandenes Wissen der Lernenden nicht ignorieren [Be98], sondern ihre Vorstellungen herausfinden und sie dann zu einer selbstständigen Problemlösung anleiten. [FS88] fordert, dass den Lernenden die Freiheit gelassen werden soll ihre eigenen Problemlöseverfahren zu entwickeln und sie nicht gezwungen werden sollen, die Strategien der Lehrkraft zu übernehmen. Das aus der Lehrerausbildung bekannte Prinzip „die Schüler dort abzuholen, wo sie sich befinden“ bedeutet im Zusammenhang mit Programmieraufgaben auch, dass die Lehrkraft nicht nur den aktuellen Stand des Lösungsversuchs betrachtet, sondern auch von den Lernenden erfragt auf welche Weise sie zu ihrem Ergebnis gelangt sind. Somit ist die Lehrkraft dann in der Lage, den Lernenden auch Hinweise zu geben wie sie die Lösung möglichst geschickt selbstständig erreichen können.

Daher ist es notwendig, mehr Wissen über die individuellen Vorgehensweisen der Lernenden bei der Lösung algorithmischer Probleme zu erhalten. Wenn die Lernumgebung die Problemlösestrategie der Lernenden bereits während des Lösungsprozesses automatisch identifizieren kann, ist es möglich (unter zusätzlicher Berücksichtigung der Qualität ihres Lösungsversuches) ihnen individualisierte Rückmeldungen zu geben. Dies unterstützt die Unabhängigkeit von der Lehrkraft während des Lernens und erhält oder fördert die Motivation der Lernenden. Außerdem wird die Kompetenz gesteigert, selbstständig Probleme zu lösen, Aufgaben zu bewältigen und sich Lerninhalte anzueignen. Zum „lifelong learning“ trägt in diesem Zusammenhang bei, dass die Lernenden sich selbstständig mit den aufkommenden Problemen auseinandersetzen und ihre eigene Problemlösestrategie wählen können. Dies führt nicht nur zu einer Verbesserung der Lern- und Programmierumgebungen sondern auch des gesamten Lehr-/Lern-Prozesses im Informatikunterricht. Zur Erreichung dieses Ziels mussten spezielle Untersuchungs- und Diagnosemodule in die eingesetzten Lernumgebungen integriert werden, damit sie die Problemlösestrategien automatisch identifizieren und individualisierte Systemrückmeldungen generieren können.

2 Vorüberlegungen

2.1 Lern- und Programmierumgebung

Um Schwierigkeiten, die neben den Problemen bei der Lösung der gestellten Aufgabe auftreten können, so gering wie möglich zu halten, musste eine Lern- und Programmierumgebung gewählt werden, die sich schnell erlernen und leicht bedienen lässt und der keine Programmiersprache mit einer komplexen Syntax zugrunde liegt. Für die ersten Studien wurde hier Kara eingesetzt. Es handelt sich dabei um eine Lernumgebung, die auf dem Prinzip der endlichen Automaten beruht. Die spezielle Terminologie der Automaten muss aber beim Lösen der hier gestellten Aufgaben nicht im Vordergrund stehen. Eine Umfrage unter Lernenden, die Kara einsetzten, ergab, dass diese Lernumgebung es ihnen ermöglichte, sich ganz auf die Problemlösung sowie die Logik und die Korrektheit ihrer Programme zu konzentrieren, ohne dabei durch die Umgebung oder Syntaxprobleme einer „echten“ Programmiersprache abgelenkt zu werden [HNR01].

2.2 Problemlösestrategien

Im Bereich der Psychologie bedeutet Problemlösung den Aufbau eines Problemraumes zwischen einem (unerwünschten) Anfangs- und einem (erwünschten) Endzustand, gefolgt von der Suche eines Weges durch den Problemraum [NS72], der durch Zwischenzustände gebildet wird ([Ma92], [CG85]). Falls man mit der Erstellung des *gesamten* Problemraumes *vor* der Suche nach einem korrekten Pfad beginnt (was nicht unbedingt notwendig ist), gibt es zwei verschiedene Möglichkeiten der Fortsetzung:

hill-climbing-Strategie

Dies ist eine vorwärts gerichtete Strategie, bei der die Lernenden in jeder Situation versuchen, eine möglichst optimale Lösung für den entsprechenden nächsten Schritt zu finden. Nachdem sie diesen Teil der Lösung überprüft haben, setzen sie mit der Suche nach dem nächsten Schritt fort. Sollten sie allerdings eine Diskrepanz zwischen ihren Erwartungen und der Situation nach ihrem Programmablauf feststellen (ein Fehlersignal im Sinne von [Oh96]), werden sie weitere Modifikationen für eine optimale Lösung dieses Schrittes vornehmen. Insgesamt verbessert sich also ihre Lösung schrittweise. Die vier in [Po73] erläuterten Schritte zur Lösung größerer Probleme, die in [BFT97] für die Informatik adaptiert wurden, finden sich hier im Kleinen beim Lösen jedes Einzelschrittes wieder.

trial-and-error-Strategie

Die Lernenden versuchen, einen Weg durch den Problemraum durch (manchmal zielloses) Ausprobieren aller verschiedenen Lösungsmöglichkeiten herauszufinden. Nach Edelman [Ed79] wird diese Strategie insbesondere dann bevorzugt, wenn es sich aus der Sicht der Lernenden um komplexe oder schwierige Aufgaben handelt. Rein zufälliges Ausprobieren lässt eine im Sinne von [CG85] gedankenlose trial-and-error-Methode entstehen, während es sich bei gezielter Auswahl der jeweils getesteten Lösungsmöglichkeit eher um eine dort als „generate-and-test-Methode“ bezeichnete Strategie handelt.

Wird der Problemraum in kleinere Teilprobleme zerlegt, so ergeben sich zwei weitere Fortsetzungsmöglichkeiten:

top-down-Strategie

Wird diese Strategie eingesetzt, identifiziert der Lernende erst sämtliche Zwischenzustände bevor er beginnt, die eigentliche Lösung zu erstellen, also den richtigen Weg durch den Problemraum zu finden. Dieses Verfahren verwendet das in der Informatik insbesondere im Bereich der Softwareentwicklung oft eingesetzte „Teile und Herrsche“-Prinzip.

bottom-up-Strategie

Hierbei lösen die Lernenden jedes einzelne Teilproblem (später dann ein Programmzweig) sobald sie es identifiziert haben, noch bevor sie versuchen, weitere Teilprobleme zu finden. Die Lösung des Gesamtproblems entsteht nach der Lösung des „letzten“ Teilproblems „automatisch“. In genau der beschriebenen Art und Weise ist diese Strategie nur einsetzbar, wenn die Teilprobleme weder verschränkt noch geschachtelt sind.

In [SL06] werden vier generelle Strategien erwähnt. Die dort als „guess-and-check“ bezeichnete Methode spiegelt sich hier sowohl in der *trial-and-error-* als auch in der *hill-climbing-Strategie* wieder, während die „sub-goal-analysis“ hier grundlegendes Prinzip der *top-down-* und der *bottom-up-Strategie* darstellt. Die von Sullivan und Lin beschriebene „ask-questions-Strategie“ ist prinzipiell ähnlich der *hill-climbing-Strategie* - hier werden die Fragen in Form von Programmausführungen an die Lernumgebung gestellt, die daraus resultierenden Situationen in der Modellwelt entsprechen den Antworten.

Nach [NS72] werden Lernende beim Problemlösen meist eine Mischung der grundlegenden Methoden einsetzen und ihre Strategie während der Lösung umfangreicherer Probleme oftmals wechseln. Einige Problemlösestrategien, die im Bereich der Psychologie erwähnt werden ([OM98], [ZG07]), kommen bei den hier beschriebenen Untersuchungen aus verschiedenen Gründen nicht in Betracht. So ist es z. B. nicht relevant, ob die Lernenden auf früher Gelerntes zurückgreifen oder bereits bekannte Lösungen abwandeln (siehe Abschnitt 2.5). Für das Erreichen des Ziels der adaptiven individualisierten Systemrückmeldungen ist es irrelevant, aus welchem Grund eine bestimmte Problemlösestrategie verwendet wird - lediglich die jeweils aktuell gezeigte Vorgehensweise ist von Belang. In beiden Fällen wird sich bei der Analyse der Untersuchungsdaten (siehe Abschnitt 2.4) kein neues Muster identifizieren lassen, da sie unabhängig davon sind, zu welchem Zeitpunkt die Versuchspersonen sich die von ihnen verwendete Problemlösestrategie angeeignet haben. Andere Problemlösestrategien wie die „Umdeutung von Werkzeugen“ sind hier offensichtlich nicht von Bedeutung. In Abhängigkeit von der jeweils gewählten Problemlösestrategie und der Qualität des zugehörigen Lösungsversuchs wurden bereits erste Überlegungen bezüglich der individualisierten Systemrückmeldungen unternommen [Ki08].

2.3 Untersuchungsmethodik

Ein Ziel der hier beschriebenen Forschungsarbeit ist es, bereits während des laufenden Lösungsprozesses mehr Erkenntnisse über die von den Lernenden eingesetzten Problemlösestrategien zu erhalten. Die bisher im Zusammenhang mit den Problemlösestrategien

von Programmieranfängern durchgeführten Studien bezogen sich vorwiegend auf Lernende in Universitäten sowie ältere Schülerinnen und Schüler. Hundhausen [Hu06] beschreibt wie der Programmierprozess von Studierenden in Abhängigkeit von der Zeit mit Hilfe von Bildschirmvideoaufnahmen analysiert werden kann. Schulte [Sc04] verwendete bei seinen Untersuchungen ähnliche Methoden, allerdings war das Thema bei ihm die objektorientierte Modellierung und er führte seine Studien in der 11. Jahrgangsstufe an Gymnasien durch. In der in diesem Artikel dargestellten Forschungsarbeit wurde versucht, die aus vorherigen Studien bekannten und z. B. in [Ch97] ausführlich beschriebenen Schwierigkeiten der Prozessbeobachtung soweit wie möglich zu vermeiden. Hierzu wurde spezielle Untersuchungs- und Diagnosesoftware, die Untersuchungsdaten in anonymisierter Form verarbeitet, entwickelt und in die Lern- und Programmierumgebung integriert. Außerdem wurden in Vorstudien mit einzelnen Testpersonen Lerner-System-Interaktionen identifiziert, die für den Problemlöseprozess relevant sind [KB08]. Es besteht die begründete Hoffnung, die Ergebnisse der hier beschriebenen Studien, insbesondere die eingesetzten Verfahren, verallgemeinern zu können, um sie auch an Universitäten mit den in Anfängerkursen im Bereich der Algorithmik eingesetzten Programmierumgebungen einzusetzen. Um die automatisierte Erkennung verschiedener Vorgehensmuster in den gesammelten Daten zu erreichen, wird ein weiteres Softwaremodul benötigt, das Muster z. B. mit Hilfe von Markov-Modellen unter Verwendung von Methoden aus der Spracherkennung prozessbegleitend identifizieren kann [KB09]. Anschließend sollen diese Muster dann verschiedenen Problemlösestrategien (siehe Abschnitt 2.2) zugeordnet und auf Grund dieser Erkenntnisse die individualisierten Systemrückmeldungen generiert werden.

2.4 Analyse individueller Vorgehensweisen

Basierend auf der oben erwähnten Kategorisierung der gesammelten Daten und deren chronologischem Verlauf während des Problemlöseprozesses können die in Abschnitt 2.2 beschriebenen Strategien automatisiert identifiziert werden. Die von der Diagnosesoftware zur Verfügung gestellten sogenannten Aktivitäts-Zeit-Diagramme (Abbildungen 2 bis 4) unterstützen die Auswertung der Daten. Die Diagramme zeigen die Verteilung der kategorisierten Lerner-System-Interaktionen jeweils einer Testperson in Abhängigkeit von der Zeit. Jede Kombination dieser Interaktionen führt zur Zuordnung der Datenmuster zu einer der vier Gruppen von „Strategie-Mustern“, die in Abbildung 1 dargestellt sind.

Falls die Lernenden zu Beginn nur *ein* Teilproblem der gestellten Aufgabe herausgreifen (zugehörige Interaktion *transition/branch*), dann dieses feiner strukturieren (Interaktion *sensor*) und danach den Zweig des Programms so vervollständigen, dass dieser Teil der Aufgabe gelöst ist (Interaktion *command*), wird das „bottom-up-Muster“ (vgl. Abbildung 3) zugeordnet. Durch eine Wiederholung dieser Abfolge erfolgt keine Änderung der Musterzuordnung, auch nicht bei *einzelnen* anderen Interaktionen wie z. B. Programmausführungen (*play*) während dieser Aktionssequenz. Schließen sich jedoch mehrere nicht in diese Folge passende Aktionen an, kommt es zu Änderungen gegebenenfalls in den Ausgangszustand „kein identifizierbares Muster“. Falls eine Erzeugung weiterer Teilzweige des Programms direkt der des ersten Teilzweigs folgt und anschließend alle zugehörigen

Feinstrukturierungen vorgenommen werden, bevor schließlich alle Kommandos eingefügt und somit die Teillösungen komplettiert werden, wird wiederum unter Berücksichtigung von Toleranzen das „top-down-Muster“ (vgl. Abbildung 2) zugeordnet.



Abbildung 1: Identifizierung von vier verschiedenen Mustern

Ähnliche Regeln für den Zusammenhang zwischen identifizierten Vorgehensmustern und zugehörigen Problemlösestrategien wurden auch für alle weiteren Strategien festgelegt. Mit Hilfe von der Diagnosesoftware zur Verfügung gestellten chronologischen Momentaufnahmen der Lösungsversuche der Lernenden und der graphischen Aufbereitung der gesammelten Daten in Aktivitäts-Zeit-Diagrammen wurden die individuellen Problemlösestrategien der Lernenden von mehreren Beobachtern untersucht. Bei der Auswertung der ca. 200 protokollierten Sitzungen ordneten zwei Informatiklehrkräfte unabhängig voneinander die gefundenen Muster jeweils den Strategien zu - hierbei herrschte gute Übereinstimmung bezüglich des Zusammenhangs zwischen den vier in Abschnitt 2.2 aufgeführten Problemlösestrategien und den in den Grafiken identifizierten Mustern.

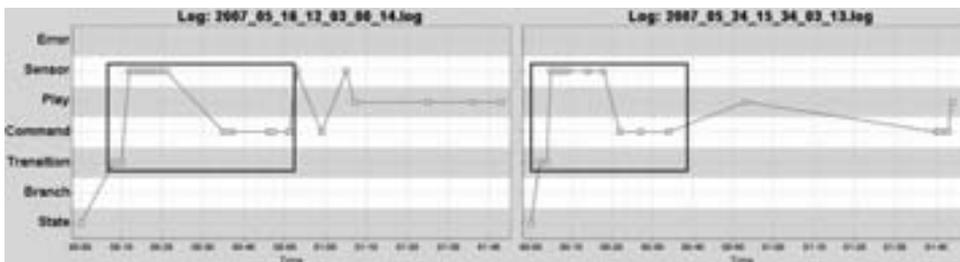


Abbildung 2: Aktivitäts-Zeit-Diagramme - Problemlösestrategie *top down*

Die Diagramme in Abbildung 2 zeigen Beispiele für Problemlöseprozesse, bei denen die Testperson jeweils zuerst alle notwendigen Zustände erstellt, danach alle Teilzweige (und Übergänge) und erst zum Schluss die zugehörigen Kommandos in den einzelnen Zweigen ergänzt (siehe Markierung). Sie zerlegt zuerst die gesamte Aufgabenstellung in einzelne Teilprobleme, deren Lösung sie erst dann durchführt, wenn der Problemraum komplett

aufgespannt ist. Diese Vorgehensweise stimmt mit der in Abschnitt 2.2 beschriebenen „top-down-Strategie“ überein. Außerhalb der Markierung ist zu erkennen, dass die Testperson zur Fertigstellung der Lösung noch ein paar einzelne Verbesserungen vornimmt. Die Diagramme in Abbildung 3 zeigen wiederum eine Strategie, bei der der gesamte Lösungsraum in einzelne Teilprobleme aufgeteilt wird. Allerdings werden hier die Kommandos in jeden einzelnen Teilzweig eingefügt bevor der nächste erzeugt und bearbeitet wird (siehe Markierung). Die endgültige Aufgabenlösung ergibt sich aus den Teillösungen nach Bearbeitung des letzten Teilzweiges. Diese Vorgehensweise entspricht der in Abschnitt 2.2 beschriebenen „bottom-up-Strategie“. Die Wiederholung des markierten Musters in Abbildung 3 verdeutlicht die schrittweise Fertigstellung der Lösungen der verschiedenen Teilprobleme. Die einzelnen Lerner-System-Interaktionen zwischen den markierten Mustern helfen dem Lernenden, seine Lösung des entsprechenden Teilzweigs zu verbessern. Wie beschrieben, werden sie bei der Identifizierung des Vorgehensmusters ignoriert.

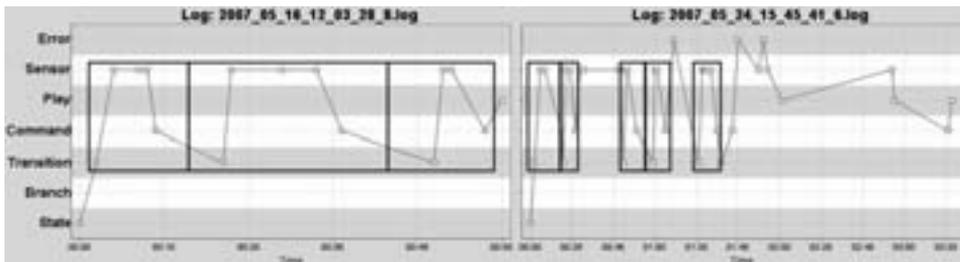


Abbildung 3: Aktivitäts-Zeit-Diagramme - Problemlösestrategie *bottom up*

Einige Lernende verwenden eine reine trial-and-error-Methode: Jeden einzelnen Schritt ihrer Lösung überprüfen sie durch Ausführung ihres Programmes (siehe Markierung in Abbildung 4 links). Die graphische Darstellung dieser Problemlösestrategie unterscheidet sich deutlich von den in Abbildung 2 und Abbildung 3 gezeigten.

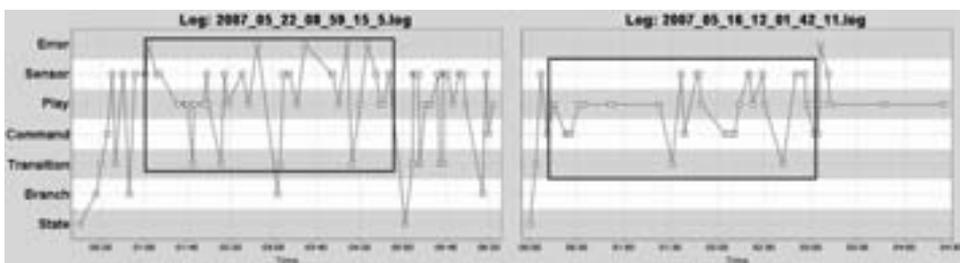


Abbildung 4: Aktivitäts-Zeit-Diagramme - links *trial and error*, rechts *hill climbing*

Zusätzlich lässt sich auch die in Abschnitt 2.2 beschriebene hill-climbing-Strategie (siehe Abbildung 4 rechts) identifizieren, bei der zwischen den Programmausführungen keine Systemfehlermeldungen auftreten, aber der Lernende für sich selbst den Stand der Lösung als unzufriedenstellend einstuft und sie dann Schritt für Schritt verbessert. Des weiteren treten Kombinationen der vier genannten Kategorien von Problemlösestrategien auf, für

die weitere von den bisherigen verschiedene Muster in den Diagrammen gefunden werden können. Ein weiteres Ergebnis ist, dass Lernende, die mit einer bestimmten Strategie begonnen haben, ein Problem zu lösen, bei einem Neuansatz der Lösungserstellung nach Auftreten von nicht zu behebenden Schwierigkeiten wiederum die gleiche Problemlösestrategie einsetzen. Eine Ausnahme bilden in dieser Hinsicht Versuchspersonen, die ihre Problemlösestrategie im Laufe des Lösungsprozesses von einer strukturierten Vorgehensweise zu einer (ziellosen) trial-and-error-Methode änderten, weil wiederholte Systemfehlermeldungen sie bei der Findung einer korrekten Lösung nicht unterstützten.

2.5 Validierung der Ergebnisse

Die Zuordnung der identifizierten Muster zu den Problemlösestrategien der Lernenden bei der Analyse individueller Vorgehensweisen soll nun in weiteren Studien verifiziert werden. Das erfordert einen alternativen Weg der Informationsgewinnung bezüglich der Problemlösestrategie der Lernenden. Dieser muss unabhängig sein von der schrittweisen Beobachtung des Problemlöseprozesses wie sie von der Untersuchungssoftware durchgeführt wird. Eine Möglichkeit ist, jede Testperson durch einen Beobachter mit der Videokamera aufzunehmen. In diesem Fall ist allerdings die Durchführung der Untersuchungen und die Analyse der gesammelten Daten sehr zeit- und arbeitsaufwändig, so dass die Methode für größere Feldstudien nicht in Frage kommt. Aus den gleichen Gründen sind auch geleitete Interviews mit jeder einzelnen Testperson im Anschluss an die Lösung der Aufgaben im Zusammenhang mit größeren Gruppen nicht realisierbar. Außerdem erfordern beide Methoden eine sehr gute Randomisierung der Auswahl der Gruppen von Testpersonen. Eine erfolgversprechende Idee ist, anhand von Fragebögen, die die Lernenden auszufüllen haben, eine Vorhersage ihres Verhaltens (hier: der eingesetzten Problemlösestrategie) zu erhalten. Dieses Ziel kann unter der Verwendung der grundlegenden Ideen von Ajzens und Fishbeins *Theorie des begründeten Handelns* und ihrer *Theorie des geplanten Verhaltens* [AF75] erreicht werden. Neben dem geringeren Zeitaufwand der Untersuchungen bietet ein Verfahren mit selbst auszufüllenden Fragebögen außerdem den Vorteil, dass der Beobachter den Lernenden nicht unerwünscht beeinflusst.

2.5.1 Ajzens Theorien

Knapp dargestellt beeinflussen nach Ajzens Theorie drei Faktoren das Verhalten: *persönliche Einstellungen* gegenüber einem bestimmten Verhalten, *subjektive Normen* in diesem Bereich und die selbst *wahrgenommene Kontrolle* über das Verhalten. Hierbei setzen sich die Einstellungen zusammen aus Erwartungen und Bewertung von Konsequenzen eines bestimmten Verhaltens, während sich die subjektiven Normen aus Normvorstellungen und der Motivation, diese zu erfüllen, ergeben. Der dritte bedeutende Faktor - die Kontrollvorstellungen - werden gebildet aus den Einstellungen hinsichtlich des Vorhandenseins von Faktoren, die die Durchführung einer bestimmten Verhaltensweise erleichtern oder behindern, und der selbst empfundenen Wichtigkeit dieser Faktoren. Die auf Ajzens Theorien bezogenen Modelle liefern zuverlässige Vorhersagen einer bestimmten Verhal-

tensweise. In den hier beschriebenen Studien entsprechen diese den verschiedenen Problemlösestrategien. Eine gute Vorhersagequote wird erreicht, wenn das Verhalten *bewusst* ausgeführt wird. Diesbezüglich wurden Ajzen und Fishbein mehrfach kritisiert, aber sie entkräfteten die Argumente in [AF80]. Ihre Modelle unterstützen nämlich die Möglichkeit, dass die Lernenden sich an früher bereits eingesetzte Verhaltensweisen erinnern, statt sich jedes Mal erneut Gedanken über Strategien zu machen. Auch in diesem Zusammenhang ist der Einsatz von Fragebögen also konsistent zur Identifizierung von Strategiemustern (siehe Abschnitt 2.4). Bei den hier beschriebenen Untersuchungen fordern alle gestellten Aufgaben die selben Kompetenzen, Überlegungen und Aktionen seitens der Lernenden. Somit sind zuverlässige Ergebnisse bezüglich der Vorhersage der von den Lernenden eingesetzten Problemlösestrategien zu erwarten.

Bestmögliche Voraussetzung für eine hohe Vorhersagequote stellt eine exakte Festlegung von Kontext und Zeitrahmen [Gu96] dar. Dies wird bei den Fragebögen durch einen einleitenden Text erreicht. So werden die gefragten Einstellungen bezüglich der Ziele der Lernenden, der Kontext und der Zeitrahmen so genau wie möglich spezifiziert. Gleiche Spezifizierungen werden für die Verhaltensweisen (Problemlösestrategien) angewendet.

2.5.2 Erstellung von Fragebögen

Die Erstellung der Fragebögen basiert auf den oben erwähnten Theorien. In einer Einleitung wird spezifiziert, dass die Erstellung der Aufgabenlösungen der Zeitrahmen und der Kontext der Befragung ist. Anschließend an diese Vorbemerkungen sollen die Lernenden einige Fragen bezüglich ihrer Ziele bei der Lösungserstellung beantworten. Mögliche Wahlantworten hierbei sind Schnelligkeit, Vollständigkeit, Korrektheit der Lösung, Verstehen des Lehrstoffs und Unabhängigkeit bei der Lösung ähnlicher Probleme. Im nächsten Teil des Fragebogens werden die Lernenden zu den drei Kategorien *Einstellungen*, *Normen* und *Kontrolle* befragt. Die Items der ersten Kategorie entsprechen den Einstellungen der Lernenden bezüglich verschiedener Aspekte der vier in Abschnitt 2.2 aufgeführten Problemlösestrategien wie in den folgenden Beispielen für die trial-and-error-Strategie zu sehen ist:

- „Meiner Ansicht nach muss ich mein Programm ständig testen, um mit Hilfe der Systemfehlermeldungen eine vollständig korrekte Lösung zu erreichen.“
- „Ich denke, um möglichst schnell eine Lösung zu finden, ist es notwendig alle Möglichkeiten auszuprobieren.“
- „Nach meiner Meinung ist das Ausprobieren aller Möglichkeiten der einfachste Weg, um eine korrekte Lösung zu erreichen.“

Die Lernenden sollen jedes Item auf einer Skala von „stimme überhaupt nicht zu“ bis „stimme völlig zu“ einordnen. Für jede der Problemlösestrategien existieren vier Items in diesem Teil der Fragebögen. Zusätzlich gibt es einige Fragen hinsichtlich der Bedeutung einzelner Faktoren in diesen Items für die Lernenden. In gleicher Weise sollen die Lernenden Fragen bezüglich ihrer subjektiven Normen im Bereich des Problemlöseprozesses beantworten. Für Jugendliche im Alter zwischen 12 und 14 Jahren werden Regeln und Normen von Eltern, Lehrkräften und Klassenkameraden gesetzt. Normen aus dem Elternhaus rufen oft Widerspruch hervor. Deshalb beschränken sich die Fragebögen auf solche,

die durch Lehrkräfte und Mitschüler bestimmt werden, wie zum Beispiel „meine Klassenkameraden meinen, dass die Strukturierung einer Problemlösung reine Zeitverschwendung ist“. Aber nicht nur Items dieser die Meinung anderer betreffenden Art sind in den Fragebögen enthalten, sondern auch solche, die deskriptive Normen behandeln wie zum Beispiel „diejenigen meiner Klassenkameraden, die keine Zeit für die Strukturierung eines Problems vor dessen Lösung verwenden, erreichen am schnellsten eine Lösung“. Ähnlich zum ersten Fragenblock schließen sich Fragen an, die die Wichtigkeit der Normerfüllung aus Sicht der Lernenden erfragen. Die letzte Fragenkategorie enthält Items, die die von den Lernenden während des Problemlöseprozesses empfundene Kontrolle betreffen. Die Testpersonen müssen hier Items bewerten wie zum Beispiel „Meine Konzentrationsfähigkeit ist nicht stark genug, um Probleme im Ganzen zu lösen - ich muss sie in kleinere Teilprobleme zerlegen und diese eins nach dem anderen lösen.“

In jeder der drei Kategorien *Einstellungen*, *Normen* und *Kontrolle* gibt es jeweils vier Items für jede der vier oben erwähnten Problemlösestrategien. Mit den zusätzlichen oben beschriebenen Items ergeben sich somit insgesamt ungefähr 60 Fragen, die die Lernenden zu bewältigen haben. Welche Fragen in welcher Zusammenstellung und Formulierung verwendet werden, wird an Hand von Beobachtungen in Unterrichtsstunden und Interviews mit verschiedenen Lehrkräften entschieden. Anschließend werden und wurden die vollständigen Fragebögen in kleineren Voruntersuchungen getestet, von denen die ersten bereits im Sommer 2008 stattfanden.

2.5.3 Zusammenfassung

Bei der Durchführung der Untersuchungen ist lediglich der Zeitpunkt, zu dem die Lernenden die Fragebögen ausfüllen sollen, noch überdenkenswert. Sie können den Lernenden sowohl vor der Stellung der ersten Aufgabe gegeben werden als auch danach. Im ersten Fall sind die Ergebnisse echte Vorhersagen des Verhaltens. Hingegen haben die Lernenden bei der zweiten Möglichkeit ein besseres Verständnis sowohl für die Items als auch den Kontext, weil sie bereits vertraut sind mit der Lösung algorithmischer Probleme mit Hilfe der Lern- und Programmierumgebung. Insbesondere bezüglich des Bereichs der von den Lernenden empfundenen Kontrolle ihres Verhaltens haben sie dann bereits Erfahrungen, welche Möglichkeiten der Kontrolle sie besitzen und in welcher Weise diese dann tatsächlich ihren Problemlöseprozess beeinflusst [SH01]. Die Fragebögen werden ausgewertet unter Berücksichtigung der Regeln aus Ajzens Theorien. Dabei werden Erwartungswert-Modelle verwendet, um mit Hilfe der Auswertung von Befragungen Vorhersagen hinsichtlich des Verhaltens der Testpersonen zu erhalten. Sich bei der Auswertung ergebende geringe Werte führen zur Ablehnung der Annahme, dass der Lernende ein bestimmtes Verhalten zeigen wird; höhere Werte bedeuten eine erhöhte Wahrscheinlichkeit des Auftretens eines bestimmten Verhaltens. Auf diese Weise ergibt sich eine Vorhersage, welche Problemlösestrategie von den Lernenden jeweils eingesetzt wird ohne dazu jeden einzelnen Schritt des Problemlöseprozesses beobachten zu müssen. Diese Vorhersagen werden dann verglichen mit den Ergebnissen aus den in Abschnitt 2.4 beschriebenen Untersuchungen. Bei einer Übereinstimmung der Aussagen ist dann die im gleichen Abschnitt beschriebene Zuordnung von Mustern aus den Beobachtungsdaten zu den Problemlösestrategien bestätigt.

3 Ausblick

Die Ergebnisse der in Abschnitt 2.4 beschriebenen Fallstudien zeigen, dass es möglich ist, verschiedene Problemlösestrategien mit Hilfe der entwickelten Untersuchungsinstrumente in Form von Mustern in den von der Untersuchungssoftware gesammelten Daten zu identifizieren. Ein nächster Schritt ist, in weiteren Untersuchungen die Zuordnung von diesen Mustern zu den Problemlösestrategien mit Hilfe der Auswertung von Fragebögen zu verifizieren. Die Fragebögen werden basierend auf den Ideen von Ajzens Theorie des begründeten Handelns erstellt. Unter Berücksichtigung der Ergebnisse der Fragebögen kann dann die Kategorisierung der Strategien verfeinert werden. Durch Integration der Fragebögen in die Diagnosesoftware kann jeder einzelne eindeutig einem speziellen Aktivitäts-Zeit-Diagramm und damit einem bestimmten Strategie-Muster zugeordnet werden. Mit Hilfe von Mustererkennungsmethoden wird eine Software entwickelt, die die Muster automatisiert identifizieren kann. Hierbei kommen Markow-Modelle, die insbesondere in der Spracherkennung häufig für ähnliche Probleme verwendet werden, zum Einsatz. Auch dieses Modul wird schließlich zusätzlich in die Lernumgebung integriert. Damit ist dann die Voraussetzung geschaffen, die Systemrückmeldungen der Lern- und Programmierumgebung an die individuellen Problemlösestrategien der Lernenden zu adaptieren.

Literatur

- [AF75] Ajzen, I., Fishbein, M.: Belief, attitude, intention, and behavior: An introduction to theory and research. Reading, MA: Addison-Wesley, 1975 (<http://people.umass.edu/ajzen/f&a1975.html>).
- [AF80] Ajzen, I., Fishbein, M.: Understanding attitudes and predicting social behavior. Englewood Cliffs, NJ: Prentice-Hall, 1980.
- [BFT97] Barnes, D. J., Fincher, S., Thomson, S.: Introductory Problem Solving in Computer Science. In 5th Annual Conference on the Teaching of Computing. Daughton G., Magee P., (eds.), Centre for Teaching Computing, Dublin City University, Ireland, 36-39, 1997.
- [Be98] Ben-Ari, M.: Constructivism in computer science education. SIGCSE Bull. 30, 1 (Mar. 1998), 257-261, 1998 (DOI=<http://doi.acm.org/10.1145/274790.274308>).
- [Ch97] Chi, M. T. H.: Quantifying Qualitative Analyses of Verbal Data: A Practical Guide. The Journal of the Learning Sciences, 6(3), 271-315, 1997.
- [CG85] Chi, M. T. H., Glaser, R.: Problem solving ability. In Human Abilities: An Information-Processing Approach. Sternberg R. (ed.). W. H. Freeman & Co, San Francisco, CA. 227-257, 1985.
- [Ed79] Edelman, W.: Einführung in die Lernpsychologie. Kösel, München, 1979.
- [FS88] Felder, R. M., Silverman L. K.: Learning styles and teaching styles in engineering education. Engineering Education, 78(7), 674-681, 1988.
- [Gu96] Güttler, P. O.: Sozialpsychologie (2. Auflage). R. Oldenbourg Verlag, München, 1996.
- [HNR01] Hartmann, W., Nievergelt, J., Reichert, R.: Kara, finite state machines, and the case for programming as part of general education. In Proceedings of the IEEE 2001 Symposium on Human Centric Computing Languages and Environments (Stresa, Italy, September 05-07, 2001). HCC'01. ACM Press, New York, NY, 135-141, 2001 (DOI=<http://doi.ieeecomputersociety.org/10.1109/HCC.2001.995251>).

- [Hu06] Hundhausen, C. D.: A Methodology for Analyzing the Temporal Evolution of Novice Programs Based on Semantic Components. In Proceedings of the 2006 International Workshop on Computing Education Research. (University of Kent, Canterbury, UK, September 9-10, 2006) ICER '06. ACM Press, New York, NY, 59-71, 2006.
- [KB08] Kiesmüller, U., Brinda, T.: Diagnosing problem solving strategies of programming novices in secondary education automatically? In Proceedings of the Joint Open and Working IFIP Conference on ICT and Learning for the Net Generation (LYICT 2008), Kuala Lumpur, Malaysia, 7-10 July), IFIP (Hrsg.), 2008.
- [KB09] Kiesmüller, U., Brinda, T.: Automatically Identifying Learners' Problem Solving Strategies In-Process Solving Algorithmic Problems. In Proceedings of the 14th Annual Conference on Innovation and Technology in Computer Science Education (Paris, France, 3-8 July, 2009). ITiCSE 2009. ACM Press, New York, NY, im Druck, 2009.
- [Ki08] Kiesmüller, U.: Diagnosing Learners' Problem Solving Strategies Using Learning Environments with Algorithmic Problems in Secondary Education. In Pre-Proceedings of the 8th KOLI CALLING INTERNATIONAL CONFERENCE ON COMPUTING EDUCATION RESEARCH (Koli, Finland, 13.-16.11.), Malmi, L., Pears, A. (eds.), 12-20, 2008.
- [Ma04] Maloney, J. et al.: Scratch: A Sneak Preview. In Second International Conference on Creating, Connecting and Collaborating through Computing. (Keihanna-Plaza, Kyoto, Japan, January 29-30, 2004) C5'04. IEEE Computer Society, Los Alamitos, CA, 104-109, 2004 (DOI=<http://doi.ieeecomputersociety.org/10.1109/C5.2004.1314376>).
- [Ma92] Mayer, R. E.: Thinking, problem solving, cognition (2nd edition). W. H. Freeman and Company, New York, NY, 1992.
- [NS72] Newell, A., Simon, H. A.: Human Problem Solving. Prentice-Hall, Englewood Cliffs, NJ, 1972.
- [OM98] Oerter, R., Montada L. (Hrsg.): Entwicklungspsychologie, 4. Auflage. Psychologie Verlags Union, Weinheim, 1998.
- [Oh96] Ohlsson, S.: Learning from performance errors. Psychological Review 103(2), 241-262, 1996.
- [Pa94] Pattis, R. E.: Karel the Robot: A Gentle Introduction to the Art of Programming, 2nd Edition. John Wiley & Sons, Inc., New York, NY, 1994.
- [Po73] Polya, G.: How To Solve It - A New Aspect of Mathematical Method. Princeton University Press, Princeton, NJ, 1973.
- [Re03] Reichert, R.: Theory of Computation as a Vehicle for Teaching Fundamental Concepts of Computer Science. Doctoral Thesis. No. 15035. ETH Zurich, 2003 (<http://e-collection.ethbib.ethz.ch/show?type=diss&nr=15035>).
- [Sc04] Schulte, C.: Empirical Studies as a tool to improve teaching concepts. In Informatics and student assessment. Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics. Magenheim, J., Schubert, S. (eds.). Köllen, Bonn. 135-144, 2004.
- [Sc97] Schwill, A.: Computer science education based on fundamental ideas. In Information Technology - Supporting change through teacher education. Passey D., Samways B., (eds.). Chapman Hall, London. 285-291, 1997.
- [SH01] Stroebe, W., Hewstone, M. (eds.): Introduction to Social Psychology: A European Perspective, 3rd Edition. Blackwell Publishers, Ltd, Malden, MA, 2001.
- [SL06] Sullivan, F. R., Lin, X.: The ideal science student and problem solving. In Proceedings of the 7th International Conference on Learning Sciences (Bloomington, Indiana, June 27 - July 01, 2006). International Conference on Learning Sciences. International Society of the Learning Sciences, 737-743, 2006.
- [ZG07] Zimbardo, P. G., Gerrig, R. J.: Psychology and Life: Pearson International Edition, 18th edition. Pearson Education, Saddle River, NJ, 2007.

Eine Studie zu den Merkmalen der Aufgabenschwierigkeit am Beispiel eines Informatik-Schülerwettbewerbs Erster Teil: Aufgabenklassifizierung

Kirsten Schlüter

Didaktik der Informatik
Universität Erlangen-Nürnberg
Martensstraße 3, 91058 Erlangen
schlueter@informatik.uni-erlangen.de

Abstract: Als erster Schritt einer Studie mit dem Ziel, den Kompetenzanspruch von Informatik-Aufgaben zu modellieren, wurde im Rahmen einer Umfrage unter Informatik-Didaktikern an Schulen und Universitäten eine Klassifizierung der Aufgaben eines Informatik-Schülerwettbewerbs für die Sekundarstufe erarbeitet. In diesem Artikel werden die Auswertungsmodalitäten und die Ergebnisse beschrieben, auf deren Basis im weiteren Verlauf der Studie die relevanten Kriterien der Aufgabenschwierigkeit bestimmt und kategorisiert werden.

1 Zielsetzung

Die Erfassung und Beurteilung des theoretischen Konstruktes informatischer Kompetenz mittels Aufgaben verlangt ein Schema zur Klassifizierung der Aufgaben, das dazu geeignet ist, die charakterisierenden Merkmale der Aufgabenschwierigkeit sichtbar zu machen. Das Schema soll sich an Bildungsstandards und an bewährten Taxonomien wie der von Bloom für den Bereich kognitiver Lernziele orientieren. Es soll anwendbar sein auf den Aufgabenbestand in Schulbüchern, Wettbewerben, Lehrportalen etc. und mit überschaubarem Einarbeitungsaufwand für Lehrpersonen mit Unterrichtspraxis und einer Fachausbildung in Informatik handhabbar sein.

Das Vorhaben der Entwicklung eines solchen Kriterienschemas wird in einer dreiteiligen Studie realisiert: Aufgabenklassifizierung, empirische Überprüfung und Kategorisierung der Merkmale. Über den ersten Teil wird hier berichtet. Ein bisher theoretisch begründeter Kriterienkatalog wird im Rahmen einer Umfrage unter Didaktikern an Schulen und Universitäten zur Klassifizierung eines Satzes von Informatikaufgaben herangezogen. Die Aufgabenklassifizierung ist der Ausgangspunkt für die empirische Überprüfung der Kriterien im zweiten Teil. Dazu werden die Aufgaben durch Schüler der Sekundarstufe bearbeitet und ihre empirische Schwierigkeit wird ermittelt. Im Abgleich der Resultate mit der Klassifizierung soll der Einfluss der Kriterien auf die Aufgabenschwierigkeit untersucht werden. Der dritte Teil umfasst die Anwendung kategorisierender statistischer Verfahren zur Auswahl und Gruppierung der relevanten Merkmale.

1.1 Ausgangssituation

Die Situation der Schulinformatik in Deutschland stellt sich auf der einen Seite so dar, dass Informatikunterricht noch längst nicht in allen Bundesländern und allen Schularten angeboten wird oder sogar verpflichtend ist. Bildungsstudien wie PISA konzentrieren sich auf die Kernfächer. Instrumente der Leistungserhebung wie etwa die zentralen Jahrgangsstufentests in Bayern in den Fächern Mathematik, Deutsch und Englisch gibt es nicht, entsprechend liegen keine Daten vor. Eine Aufgabekultur wie in der Mathematik konnte sich noch nicht entwickeln. Auf der anderen Seite steht das Engagement vieler Lehrpersonen mit Expertise im Fach Informatik und der Informatik-Didaktiker an den Universitäten, die eine Vielzahl von Unterrichtsbeispielen veröffentlichen und Aufgaben zum Lernen, Üben, Vertiefen und Prüfen entwickeln. Mit dem Ziel, das Schulfach Informatik in der Sekundarstufe zu stärken und zu konsolidieren, wurden im Jahr 2008 in der Gesellschaft für Informatik Bildungsstandards verabschiedet [GI08].

1.2 Taxonomien, Kompetenzmodelle, Ordnungsschemata

Puhlmann und Friedrich stellen dar, wie die Bildungsstandards der Informatik zur Beurteilung von Aufgaben für Unterricht und Test im Hinblick auf die Prozess- und Inhaltskompetenzen anwendbar sind [PF07]. Kompetenzmodelle, die sich auf einzelne Inhaltsbereiche beziehen, werden in jüngeren Arbeiten vorgestellt, z. B. von Kohl und Fothe für Algorithmen [KF07] und von Freischlad für Internetworking [Fr08]. Von Starr, Manaris und Stalvey wird ein Ansatz beschrieben, Blooms originale Taxonomie zur Curriculumsauslegung und Aufgabengestaltung heranzuziehen [St08]; Fuller et al. wenden eine zweidimensionale Version entlang der Dimensionen Interpreting und Producing an [Fu08]; Lahtinen beschreibt eine Typisierung von Programmieranfängern nach Blooms Lernzielstufen [La07]. Biggs und Colins stellen die SOLO Taxonomie vor (Prestructural, Unistructural, Multistructural, Relational) [BC82], die von Sheard et al. zur Klassifizierung von Prüfungsantworten eingesetzt wird [Sh08]. Bower schlägt eine zehnstufige Taxonomie (Declarative, Comprehension, Debugging, Prediction, Provide-an-example, Provide-a-model, Evaluate, Meet-a-design-specification, Solve-a-problem und Self-reflect tasks) zur Kategorisierung von Aufgaben vor [Bo08]. Curriculumsbedingt liegt der Fokus der internationalen Ansätze auf dem Lerninhalt Programmierung. In Deutschland hat Brinda für die strukturierte Gestaltung von Aufgaben zur objektorientierten Modellierung eine Klassifizierung nach Fachkern, Gegenstand und Aufgabentyp mit einer Einordnung der Aufgabentypen in Blooms Taxonomie entwickelt [Br04]. Steinert präsentiert ein curriculares Ordnungsschema zur Lernerfolgsanalyse, die Konstruktion von Lernzielgraphen aus Aufgaben und Musterlösungen [St07].

1.3 Spezifische Schwierigkeitsmerkmale von Informatikaufgaben

Allen Schemata ist gemeinsam, dass sie Curriculumsinhalte, Aufgaben oder Antworten klassifizieren oder Lerner kategorisieren. Sie sind nicht dazu geeignet, die Schwierigkeit von Aufgaben herauszuarbeiten. Das Augenmerk dieser Arbeit liegt auf den Kriterien, die die Schwierigkeit von Aufgaben und damit ihren Kompetenzanspruch bestimmen. Bisher wurden die charakterisierenden Merkmale Erfahrungsweltnähe, Abstraktionsgrad,

Komplexität, Formalisierungsgrad, Redundanz, Anforderungsbereich, Prozessbereich, kognitive Lernzielstufe und Art des Wissens identifiziert [Sc08]. Ausgehend von der Annahme, dass das komplexe Gefüge informatischer Kompetenz sich durch einfache Merkmale modellieren lässt, soll langfristig ein Klassifizierungsschema entwickelt werden, das im Abgleich mit dem daraus erwachsenden Kompetenzmodell dazu dient, Aufgaben anhand einfacher Kriterien nach ihrem Kompetenzanspruch zu klassifizieren.

2 Aufgabenklassifizierung: Durchführung

Zur empirischen Überprüfung der Merkmale werden die Aufgaben des Schülerwettbewerbs Informatik-Biber [Po09] verwendet – bewusst „echte“ statt eigens auf die Merkmale zugeschnittene Aufgaben und bewusst „alle“ statt ausgewählte Aufgaben. Der Informatik-Biber, eine Kooperation des Bundeswettbewerbs Informatik und der Arbeitsgruppe Didaktik der Informatik der Universität Münster, ist ein jährlich ausgerichteter Online-Wettbewerb, der sich an Schüler der Jahrgangsstufen 5 bis 7, 8 bis 10 und ab 11 richtet und keine Vorkenntnisse in Informatik voraussetzt. Die Aufgaben sind kurz und schnell zu erfassen. Sie entstammen den Themenbereichen, die für die Informatik als Schulfach relevant sind, und sollen informatische Denkweisen und Problembearbeitung erfordern, aber kein spezifisches Fachwissen. Die folgende Beispielaufgabe „Biber und Bisons“ wurde im Wettbewerb 2007 in den Jahrgangsstufen 8 bis 10 und ab 11 gestellt.

Biber sagen immer die Wahrheit, und Bisons lügen immer. Im Biber-und-Bison-Zeltlager wohnen insgesamt zehn Tiere. Ein blinder Maulwurf kommt vorbei und möchte wissen, wie viele Biber und wie viele Bisons anwesend sind. Darum fragt er jedes Tier: "Wie viele Biber gibt es hier?" Die zehn Antworten sind: 3, 4, 1, 4, 1, 1, 3, 4, 3, 2. Jetzt weiß der blinde Maulwurf genau Bescheid! Du auch?

Wie viele Biber sind im Biber-und-Bison-Zeltlager? A) 1 B) 2 C) 3 D) 4

2.1 Vorgehensweise

Gestützt durch eine umfangreiche Expertenumfrage werden im ersten Teil der Studie die Wettbewerbsaufgaben klassifiziert, über die Ergebnisse wird hier berichtet. (Die Klassifizierung wird allerdings noch nicht zur Kategorisierung der Aufgaben herangezogen, vielmehr soll sie im zweiten Schritt anhand der Bearbeitungsergebnisse der Wettbewerbsteilnehmer überprüft werden.) Um eine möglichst objektive Einordnung in das Kriterienschema zu erreichen, lag jede Aufgabe mehreren Personen mit Expertise in der Informatik-Didaktik zur Einschätzung vor. Weil die Einarbeitung in die Kriterien und die sorgfältige Einschätzung je Aufgabe einen beträchtlichen Zeitaufwand erfordern, wurden die 29 Aufgaben in Arbeitseinheiten zu je fünf Aufgaben aufgeteilt. Den Befragten lagen zur Orientierung außerdem drei Beispielaufgaben vor, deren Einordnung in einer Vorstudie innerhalb der Arbeitsgruppe Didaktik der Informatik der Universität Erlangen-Nürnberg abgestimmt wurde, sowie die folgende Spezifikation der Kriterien.¹

¹ Unter <http://ddi.informatik.uni-erlangen.de/research/kompetenzmodell/> stehen die versendeten Unterlagen (alle Aufgabenprofile, die Beispielpprofile, Hinweise zu den Kriterien) zur Verfügung.

Unter der **Erfahrungsweltnähe (EN)** einer Aufgabe soll verstanden werden, wie nah sie dem persönlichen Erleben der Schüler ist. Dabei ist der Aufgabenkontext zu betrachten, nicht der Fachinhalt. Zum Beispiel könnte der Fachinhalt „Scheduling“ in unterschiedlich lebensweltnahe Kontexte eingebettet werden: Der Aufgabenkontext „Stundenplan“ ist dem Schüleralltag zuzuordnen, das heißt *nah* (=1) zur Erfahrungswelt. Der Aufgabenkontext „Fahrplan“ ist nicht unbedingt täglich erlebt, aber jederzeit erlebbar durch die Schüler, das heißt, er weist *mittlere* (=2) Entfernung zur Erfahrungswelt auf. Der Aufgabenkontext „Produktionsplanung“ ist im Allgemeinen nicht erlebbar durch die Schüler, jedoch vorstellbar, das heißt *entfernt* (=3) von der Erfahrungswelt.

Unter dem **Abstraktionsgrad (AG)** einer Aufgabe soll der Grad der Loslösung vom konkreten Beispiel verstanden werden: Lautet die Aufgabe, einen Weg in einem gegebenen Netzwerk zu suchen, so handelt es sich um einen konkreten Fall und die Aufgabe ist vom Abstraktionsgrad *konkret* (=1). Lautet die Aufgabe, mögliche Knotenausfälle in einem Netzwerk durchzuspielen, so handelt es sich um die Betrachtung gedachter Fälle. Die Aufgabe ist von *mittlerem* (=2) Abstraktionsgrad. Gilt es, einen Algorithmus zur Wegsuche in Netzwerken zu finden, so handelt es sich um ein allgemeines Verfahren, losgelöst vom Einzelfall. Die Aufgabe ist *abstrakt* (=3).

Die **Komplexität (KG)** ergibt sich aus der Anzahl der Fälle, Regeln oder Umformungen, die die Größe des Lösungsraums bestimmt. Zum Beispiel könnte eine Aufgabe im Kontext Schachspiel unterschiedlich komplex gestaltet werden: Ist ein einzelner Spielzug einer Figur zu betrachten, gilt die Aufgabe als *einfach* (=1). Sind es die Züge mehrerer Figuren, ist die Komplexität *mittel* (=2). Ist eine Folge von Zügen mehrerer Figuren zu betrachten, gilt die Aufgabe als *komplex* (=3).

Der **Formalisierungsgrad (FG)** ist ein Merkmal der Informationsdarstellung. Ein Text – „Von drei Personen ist A die größte, B die kleinste und C die mittlere.“ – oder ein Bild ist *informell* (=1). Die symbolische Repräsentation weist einen *mittleren* (=2) Formalisierungsgrad auf. Die modellhafte Darstellung, etwa als Baum oder Struktogramm ist *formal* (=3).



Die **Redundanz (RG)** ist ebenfalls ein Merkmal der Informationsdarstellung. Wird die gleiche Information mehrfach dargestellt, etwa als Text – „Von drei Personen ist A die größte, B die kleinste und C die mittlere.“ – und zusätzlich als Bild, so ist die Informationsdarstellung in der Aufgabe *redundant* (=1), sonst *nicht redundant* (=2).

Der **Anforderungsbereich (AB)** bezieht sich auf die geforderte Aktion. Drei Anforderungsbereiche werden in den Einheitlichen Prüfungsanforderungen in der Abiturprüfung Informatik [Ku04] unterschieden: *Wiedergabe* (=1), z. B. das Dualsystem beschreiben, *Anwendung* (=2), z. B. eine Zahl aus dem Dezimalsystem in das Dualsystem übertragen, und *Problemlösung* (=3), z. B. ein Verfahren zur Addition dualer Zahlen entwickeln.

Auch der **Prozessbereich (PB)** bezieht sich auf die geforderte Aktion. Fünf Prozessbereiche werden in den Bildungsstandards Informatik [GI08] unterschieden: *Modellieren und Implementieren* (=1), *Begründen und Bewerten* (=2), *Strukturieren und Vernetzen* (=3), *Kommunizieren und Kooperieren* (=4), *Darstellen und Interpretieren* (=5).

Sechs **kognitive Lernzielstufen (LS)** werden nach der zweidimensionalen Taxonomie von Anderson, Krathwohl et al. [AK01] in der ersten Dimension unterschieden: *Erinnern* (=1), *Verstehen* (=2), *Anwenden* (=3), *Analysieren* (=4), *Bewerten* (=5) und *Erschaffen* (=6).

In der zweiten Dimension der Taxonomie werden vier **Arten des Wissens (AW)** unterschieden: *Fakten* (=1), *Konzepte* (=2), *Prozeduren* (=3) und *Metakognition* (=4).

Die Kommunikation mit den Experten wurde ausschließlich schriftlich geführt. Die erste Kontaktaufnahme fand per E-Mail statt. Die Fragebögen lagen auf Papier vor und sowohl Versand als auch Rücklauf erfolgten per Post. Von den 45 Befragten sind neun weiblich. 21 unterrichten in der Sekundarstufe, davon 14 auch an der Universität. 22 sind nun an der Universität, zwei sind außerhalb der Lehre tätig (Abb. 1).

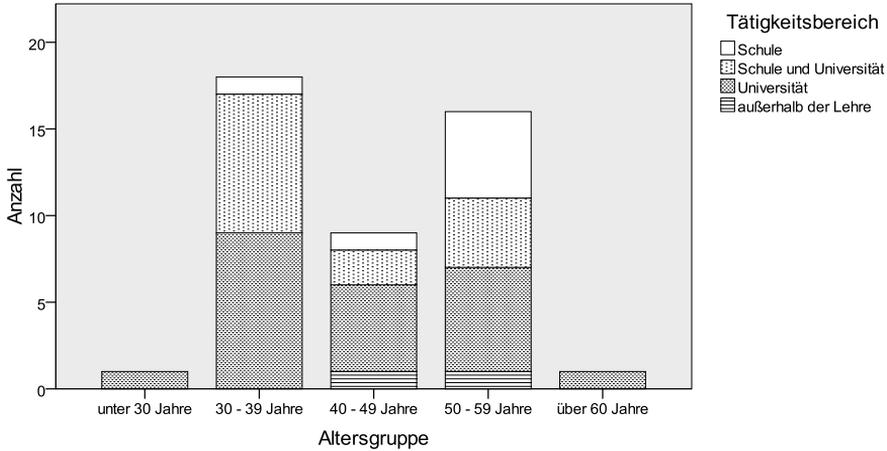


Abbildung 1: Altersgruppe und Tätigkeitsbereich der 45 Befragten

2.2 Datengrundlage

Der Studie liegen die anonymisierten Teilnahmedaten des Informatik-Bibers 2007 zu Grunde. Das Datenmaterial umfasst Jahrgangsstufe, Geschlecht, Bundesland und die Bearbeitungsergebnisse je Aufgabe für 21.802 Teilnehmer. Insgesamt wurden 29 Aufgaben aus allen Inhaltsbereichen laut Bildungsstandards Informatik gestellt (Abb. 2).

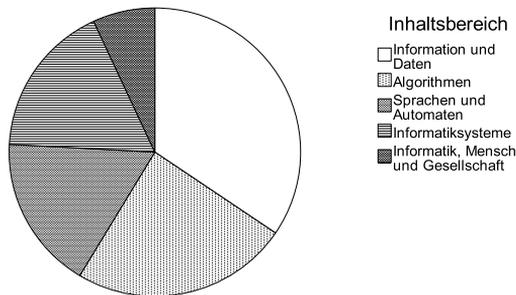


Abbildung 2: Inhaltsbereiche der Aufgaben des Informatik-Bibers 2007

Stärkster Inhaltsbereich ist *Information und Daten* mit zehn Aufgaben, davon zwei zum Thema Datenstrukturen, vier zu Graphen, vier zu Codierung. Auf *Algorithmen* entfallen sieben Aufgaben, auf *Sprachen und Automaten* fünf Aufgaben, eine zu Sprachen, eine zu Automaten, drei zum Logischen Schließen. (Die Bildungsstandards beziehen das Thema Logisches Schließen nicht explizit ein. Als formales Kalkül wurde es hier den *Sprachen*

zugeordnet.) Von den fünf Aufgaben des Inhaltsbereichs *Informatiksysteme* betreffen vier das Thema Anwenderwissen, eine das Thema Datenbanken. *Informatik, Mensch und Gesellschaft* ist mit nur zwei Aufgaben vertreten, beide zum Benutzerverhalten. Die Aufgaben wurden teilweise in mehreren Jahrgangsstufen gestellt. Aus Gründen der inhaltlichen Ausgewogenheit kam es auch vor, dass Aufgaben zum Beispiel in den Stufen 5 bis 7 und ab 11, nicht aber in 8 bis 10 gestellt wurden.

3 Aufgabenklassifizierung: Auswertungsmodalitäten

Die aufgabenweise Sichtung der 225 Aufgabenprofile ergab, dass je Aufgabe zwischen drei und siebzehn Einschätzungen vorliegen. Diese Spanne ist breit, nicht etwa weil sich die Aufgaben so unterschiedlich einschätzbar zeigten, sondern weil Aufgaben, die im Wettbewerb in mehreren Jahrgangsstufen gestellt wurden, entsprechend auch mehreren Experten zur Einschätzung vorlagen, und weil der Rücklauf nicht vollständig war, nur 45 von 54 versendeten Fragebogensätzen wurden zurückgesendet. Zunächst werden die Häufigkeitsverteilungen der Bewertungen je Merkmal betrachtet (Abb. 3).

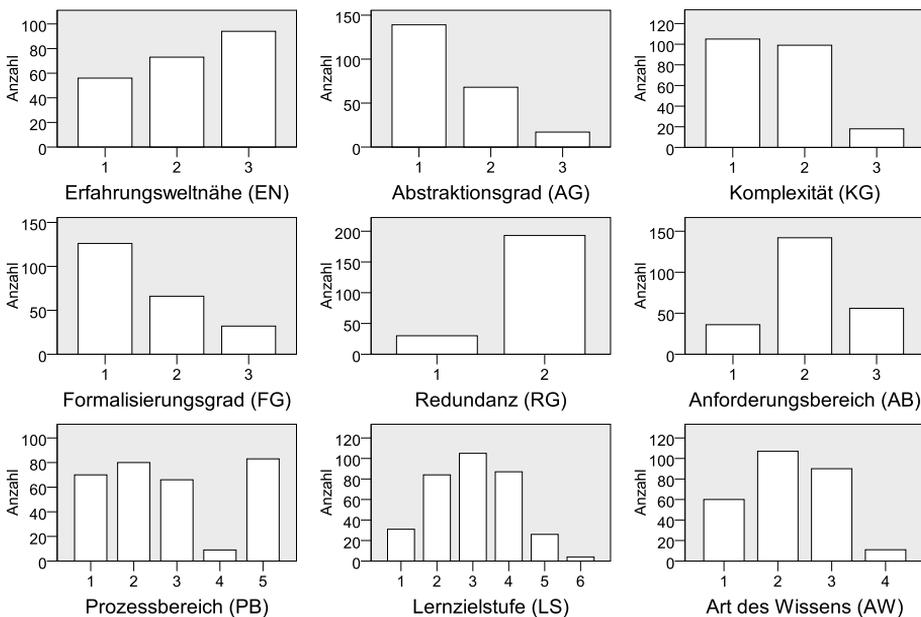


Abbildung 3: Häufigkeitsverteilungen der Merkmale EN bis AW auf der Basis von 225 Experten-Einschätzungen. Bei den Merkmalen AB bis AW waren Mehrfachantworten möglich.

Die Verteilung der Erfahrungsweltnähe ist steigend, die meisten Einschätzungen lauten *entfernt* von der Erfahrungswelt. Die Verteilungen der Merkmale Abstraktionsgrad, Komplexität und Formalisierungsgrad sind fallend, besonders *abstrakt* und *komplex* werden selten genannt. Das trifft auch auf *redundant* zu. Beim Anforderungsbereich fällt auf, dass der mit Abstand größte Anteil auf *Anwendung* entfällt. Die Prozessbereiche

werden gleichermaßen genannt, nur *Kommunizieren und Kooperieren* wird im Quiz-Format kaum angesprochen. Die Einschätzungen der Lernzielstufe sind glockenförmig um den Schwerpunkt *Anwenden* verteilt. Besonders *Erschaffen* ist dem Wettbewerbsformat Multiple-Choice entsprechend selten, wie auch die Wissensart *Metakognition*.

Um nun aus den Einzelbewertungen eine möglichst objektive Gesamtbewertung zu gewinnen, werden die Häufigkeitsverteilungen der Merkmale je Aufgabe ausgewertet. Tabelle 1 zeigt das Gesamtergebnis der Aufgabenklassifizierung.

Tabelle 1: Aufgabenklassifizierung nach den charakterisierenden Merkmalen, schattiert nach der Ausprägung. Durchschnittliche Übereinstimmung und Standardabweichung. (***) 1, 2, 5)

	EN 1-3	AG 1-3	KG 1-3	FG 1-3	RG 1-2	AB 1-3	PB 1-5	LS 1-6	AW 1-4
Bibers Geheimcode	2	1	2	1	2	2	3	3	3
Biberzahlen	3	2	2	2	2	2	5	3	3
Verwandlung	3	2	2	1	2	3	2 5	4	3
Zahlenreihe	3	1	1	1	2	2	1	3	3
Falschgeld	2	1	2	2	2	2	5	3	3
Dateisuche	2	2	2	2	2	2	3	2 3	3
Private E-Mail	1	1	1	1	2	1	2 4	1	1
Wetter	1	1	1	1	2	2	2 3	2 3	2
Schnitzeljagd	3	1	2	3	2	2	2	3 4	3
Endlos-Schleife	3	2	2	2	2	2	1 5	3 4	2 3
Verschlüsselung	3	2	2	2	2	2	***	3 4	2
Wertetausch	2	3	2	3	2	2	2	3	2 3
Ungeschützter Computer	1	1	1	1	2	1	2	2	1
Computervirus	1	1	1	1	2	1	2	2	1
Platzwechsel	3	2	2	2	2	3	1	3 4	3
Link	1	1	1	1	2	1	4	1	1
Primärschlüssel	3	1	1	2	2	2	2	2	1 2
Sicheres Passwort	1	1	1	1	2	2	2	5	1
Fenster schließen	1	1	1	1	2	1	5	1	1
Labyrinth	2	2	2	1	1	2 3	3 5	3 4	2 3
Binärbaum	3	1	2	3	2	2	5	3 4	2
Umparken	2	1	1	2	2	2	1 5	3	2 3
Netzwerkkabel	3	2	2	2	2	3	3 5	4	2
Morse-Code	3	1	2	2	2	2 3	3	4	3
Biber am Fluss	2	2	1	1	2	2	5	4	2 3
Dino-Ordnung	2	1	2	1	2	2	2 3	2 4	2
Links um!	2	1	1	1	2	2	1	3	3
POP und PUSH	2	1	2	1	2	2	1 5	3	2 3
Biber und Bisons	3	2	2	1	2	3	2	2 4	2
Durchschnittliche Übereinstimmung (%)	73,7	68,4	69,1	72,3	85,8	81,8	68,0	77,6	69,6
Standardabweichung (%)	13,4	20,1	16,9	19,3	18,4	14,2	15,8	17,4	16,7

Für die Merkmale EN bis RG gelten andere Auswertungsmodalitäten als für AB bis AW, bei denen Mehrfachantworten möglich waren. Als Aufgabenbewertung bezüglich EN bis RG wird der Modalwert der Bewertungsverteilung gewählt, also der meistgenannte Wert. Bei zwei benachbarten Modi wird die stärkere Ausprägung gewählt, das heißt *abstrakt*, falls von zehn Befragten je 0/5/5 *konkret/mittel/abstrakt* nennen (a). In allen anderen Fällen wird keine Gesamtbewertung ausgewiesen. Bezüglich AB bis AW ergibt sich die Bewertung aus den übereinstimmenden Nennungen von mindestens 50 % der Befragten, das heißt *Konzepte* und *Prozeduren*, falls von zehn Befragten je 2/8/5/1 die Wissensarten *Fakten/Konzepte/Prozeduren/Metakognition* nennen (b).

Als Maß für die Übereinstimmung der Einschätzungen wurde die prozentuale Nennungshäufigkeit des Modalwertes festgelegt. Das obige Beispiel (a) weist 50 % Übereinstimmung auf (0/5/5 Nennungen der Abstraktionsgrade 1-3 entsprechen bei zehn Befragten 0/50/50 %), und Beispiel (b) 80 % (2/8/5/1 Nennungen der Wissensarten 1-4 entsprechen bei zehn Befragten 20/80/50/10 %). Die durchschnittliche Übereinstimmung für ein Merkmal (Tab. 1) kann einen Hinweis auf dessen Anwendbarkeit und Handhabbarkeit geben. Die Klassifizierung hängt vom Übereinstimmungsmaß ab und wird im ersten Gang mit der Maßgabe durchgeführt, möglichst jeder Aufgabe bezüglich jedes Merkmals eine Bewertung zuzuweisen. Falls sich in der Analyse der empirischen Daten keine deutlichen Ergebnisse abzeichnen, kann zum Beispiel der Parameter Mindestübereinstimmung verändert werden, um eine strengere Klassifizierung zu gewinnen.

Die Aufgaben gehen gemäß der Klassifizierung in den zweiten Teil der Studie ein: die Beispielaufgabe „Biber und Bisons“ (Kap. 2) dient so als Testaufgabe für die Attribute *entfernt* von der Erfahrungswelt, *mittlerer* Abstraktionsgrad, *mittlere* Komplexität, *informell*, *nicht redundant*, Anforderung *Problemlösung*, Lernzielstufen *Verstehen*, *Analisieren* und *Konzeptwissen*. (Für den Prozessbereich wurde keine übereinstimmende Einordnung erzielt. Von neun Befragten wurden die Bereiche 1-5 mit den Häufigkeiten 3/3/4/0/1 genannt.²) Im Folgenden wird reflektiert, inwieweit die Aufgaben sich der Einordnung nach den Kriterien erschließen, die Kriterien auf Akzeptanz seitens der Befragten stoßen und eine übereinstimmende Aufgabenbewertung möglich ist.

3.1 Erfahrungsweltnähe

Allein nach dem Grad der Übereinstimmung beurteilt, den die Befragten erreichen, scheint das Kriterium Erfahrungsweltnähe (EN) mit einer durchschnittlichen Übereinstimmung von 73,7 % bei einer Standardabweichung von 13,4 % gut anwendbar und handhabbar zu sein – im Durchschnitt gelangen drei von vier Experten zur gleichen Bewertung. Allerdings erfüllt das Merkmal nur bedingt den Anspruch, ausschließlich aufgabenbezogen einschätzbar zu sein, zumindest in der Sekundarstufe, denn die Erfahrungswelt z. B. von Zehn- und Sechzehnjährigen oder Gymnasiasten und Berufsschülern ist unterschiedlich. Dadurch ist die Anwendbarkeit begrenzt auf Aufgaben für bekannte Adressatengruppen. Auffallend ist weiter, dass die Befragten je nach Geschlecht unter-

² Unter <http://ddi.informatik.uni-erlangen.de/research/kompetenzmodell/> stehen die Häufigkeitstabellen für alle Aufgaben und Kriterien zur Verfügung.

schiedlich bewerten (Abb.4). Aus der Sicht der männlichen Befragten ist der größte Anteil, aus der Sicht der weiblichen Befragten der geringste Anteil der Aufgaben *entfernt* (=3) von der Erfahrungswelt. Das schränkt die Handhabbarkeit des Kriteriums ein.

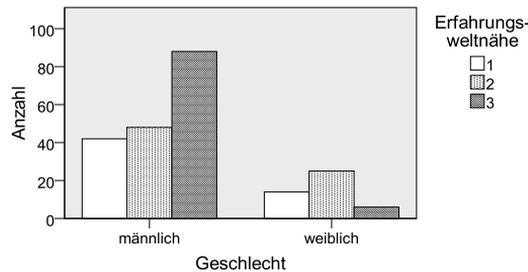


Abbildung 4: Geschlecht der Befragten und Einschätzung der Erfahrungsweltnähe

3.2 Abstraktionsgrad

Der Abstraktionsgrad (AG) hat im Merkmalsvergleich mit 68,4 % die geringste, etwa eine Zweidrittel-Übereinstimmung, bei einer relativ großen Streuung von 20,1 %. Zwar wählen die Experten bei „Zahlenreihe“, „Private E-Mail“, „Link“, „Sicheres Passwort“ zu 100 % übereinstimmend den Abstraktionsgrad *konkret*, aber bei „Biberzahlen“, „Wertetausch“, „Dateisuche“ verteilen sich die Einschätzungen zu 40/40/20 %, 20/40/40 % oder gar 40/20/40 % auf die Stufen *konkret/mittel/abstrakt*. Ein Erklärungsansatz ist, dass auf den Kontext statt auf die eigentliche Arbeitsanweisung Bezug genommen wird.

3.3 Komplexität

Die Komplexität (KG) weist mit 69,1 % eine ähnlich geringe Übereinstimmung auf wie der Abstraktionsgrad, es können aber alle Aufgaben bezüglich der Komplexität eindeutig klassifiziert werden. Zu bedenken bleibt, ob das spezifizizierte Kriterium allen Facetten des Kompetenzanspruchs gerecht wird, der aus der Aufgabenkomplexität erwächst, beansprucht doch ein größerer Lösungsraum vielfältige Kompetenzen, zum Beispiel Konzentrations- und Durchhaltevermögen oder strategisches Denken.

3.4 Formalisierungsgrad

Die Bewertung des Formalisierungsgrades (FG) erfolgte mit zufriedenstellender Übereinstimmung (72,3 %), lediglich „Dateisuche“ fand keine eindeutige Bewertung. Die relativ starke Streuung (19,3 %) weist jedoch auf eventuelle Schwächen in der Spezifikation von *informell, mittel und formal* hin. Aus den Anmerkungen der Befragten geht hervor, dass die Stufen umfassend und trennscharf sind, aber ihre Anordnung wird hinterfragt. Eine symbolische Darstellung, beispielsweise eines Automaten als Tupel, weise gegebenenfalls einen höheren Formalisierungsgrad auf als eine modellhafte Repräsentation, etwa als Graph. Erst von der empirischen Überprüfung ist Aufschluss darüber zu erwarten, wie sich der Schwierigkeitsgrad zum Formalisierungsgrad verhält.

3.5 Redundanz

Die Bewertungsverteilung der Redundanz (RG) zeigt bereits, dass die Einschätzung *nicht redundant* deutlich überwiegt (Abb. 3). Unter den gegebenen Auswertungsmodalitäten würden mit 85,8 % durchschnittlicher Übereinstimmung alle Aufgaben als *nicht redundant* klassifiziert. Bei benachbarten Modi die stärkere Ausprägung zu wählen ist hier jedoch unangemessen, weil *redundant* und *nicht redundant* nicht als benachbart, sondern als gegenteilig zu betrachten sind. Deshalb bleibt bei „Biberzahlen“, „Private E-Mail“ und „Platzwechsel“ die Frage der Redundanz unentschieden. Der vorliegende Aufgabensatz, in dem „Labyrinth“ als einzige Aufgabe *redundant* klassifiziert wird, ist nicht geeignet, den Zusammenhang zwischen Redundanz und Aufgabenschwierigkeit zu überprüfen.

3.6 Anforderungsbereich

Der Anforderungsbereich (AB) wurde von den Befragten mit großer Übereinstimmung (81,8 %) bei geringer Standardabweichung (14,2 %) eingeschätzt. Da die Wettbewerbsteilnahme keine Vorkenntnisse voraussetzt, kann das Kriterium aufgabenbezogen, das heißt unabhängig vom Adressaten, beurteilt werden. Für jede Aufgabe werden in der Vorgabe alle erforderlichen Informationen zur Verfügung gestellt, die mit Sachverstand wiederzugeben, anzuwenden oder auf deren Basis das Problem zu lösen ist, um die richtige Antwortmöglichkeit identifizieren zu können. Es besteht ein gewisser Zusammenhang zwischen Anforderungsbereich und Lernzielstufe (s. dazu Abb. 5 und Kap. 3.8).

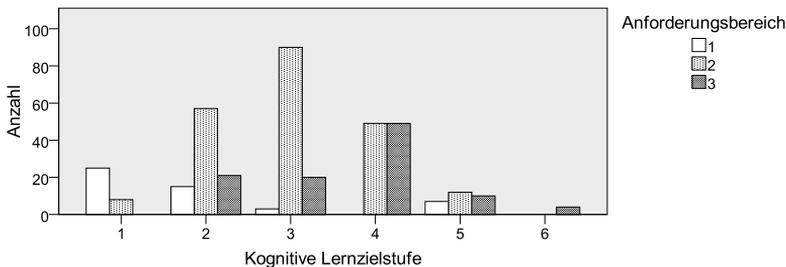


Abbildung 5: Kognitive Lernzielstufe und Anforderungsbereich

3.7 Prozessbereich

Der Prozessbereich (PB) zeigt mit 68 % die geringste Übereinstimmung. Elf der 45 Befragten vermerken, die Einschätzung dieses Merkmals sei mit der größten Unsicherheit erfolgt. Drei Experten geben an, sie hätten regelmäßig rein nach dem Ausschlussprinzip *Modellieren und Implementieren* gewählt. Nichtsdestoweniger zeigt die Bewertungsverteilung, dass die Prozessbereiche, bis auf *Kommunizieren und Kooperieren*, gleichermaßen in Anspruch genommen werden. Ob die resultierende Klassifizierung geeignet ist, einen Einfluss des Unterscheidungsmerkmals Prozessbereich auf die Aufgabenschwierigkeit festzustellen, wird im zweiten Teil der Studie untersucht.

3.8 Lernzielstufe

Die Einordnung der Aufgaben in die Lernzieltaxonomie (LS) erfolgt mit 77,6 % Übereinstimmung. Da den kognitiven Lernzielstufen so wie den Anforderungsbereichen die Taxonomie von Bloom zugrunde liegt, gibt es Überschneidungen in der Bewertung. Die gruppierte Häufigkeitsverteilung (Abb. 5) lässt erkennen, dass beispielsweise die Lernzielstufe *Anwenden* (=3) häufig mit dem Anforderungsbereich *Anwendung* (=2) zusammenfällt, die Bewertungen jedoch nicht zur Deckung gebracht werden können, da die Lernzielstufe *Analysieren* (=4) genau so häufig mit der Anforderung *Anwendung* (=2) zusammenfällt wie mit *Problemlösung* (=3). Welches der beiden Merkmale besser geeignet ist, die Aufgabenschwierigkeit abzubilden, ist eine Fragestellung der geplanten empirischen Untersuchung.

3.9 Art des Wissens

Die Wissensart (AW) wird mit relativ geringer Übereinstimmung von 69,6 % eingeschätzt. Die Aufgaben „Bibers Geheimcode“ und „Links um!“ können nicht eingeordnet werden. Der Wissensart *Metakognition* kann im Gesamtergebnis keine Aufgabe zugeordnet werden (die höchste Bewertung erreicht die Aufgabe „Biber und Bisons“, die den Wissensarten *Fakten/Konzepte/Prozeduren/Metakognition* von neun Befragten mit 2/6/4/3 Nennungen, das entspricht 22/67/44/33 %, zugeordnet wurde).

4 Zusammenfassung und Ausblick

Das Kriterienschema erweist sich als gut handhabbar und wird von den Befragten akzeptiert, wesentliche Kritikpunkte werden in Kapitel 3 diskutiert. Die Aufgaben erschließen sich der Bewertung: lediglich einzelne Aufgaben können bezüglich eines der Kriterien nicht eingeordnet werden, und nur die Lernzielstufe *Erschaffen* sowie die Wissensart *Metakognition* werden im Gesamtergebnis gar nicht erreicht. Bis auf die Erfahrungsweltnähe sind die Einschätzungen unabhängig von Geschlecht, Altersgruppe und Tätigkeitsbereich der Befragten. Einzig die Vertrautheit mit den Aufgaben wirkt sich aus, indem in der Gruppe der Befragten, die mit den Aufgaben des Informatik-Bibers noch nicht vertraut waren, sämtliche Merkmale eine größere Varianz bei gleicher Zentraltendenz aufweisen als in der Gruppe derer, die mit den Aufgaben schon vertraut waren.

Im ersten Teil der Studie wurde eine Klassifizierung der Aufgaben des Schülerwettbewerbs Informatik-Biber nach den charakterisierenden Merkmalen der Schwierigkeit erarbeitet. Darüber wurde in diesem Artikel berichtet. Im zweiten Teil, der noch nicht abgeschlossen ist, werden die Teilnahmedaten des Wettbewerbs, insbesondere die Bearbeitungserfolge, nach der Aufgabenklassifizierung aufgeschlüsselt. Das Ziel ist, zu überprüfen, ob und in welchem Maße die klassifizierenden Merkmale relevante Kriterien der Schwierigkeit von Aufgaben darstellen. Im abschließenden dritten Teil der Studie soll mittels kategorisierender statistischer Analyseverfahren das Schwierigkeitsgefüge als Modell des Kompetenzanspruchs sichtbar gemacht werden.

Danksagung

Danke an die Expertinnen und Experten der Informatik-Didaktik für die sorgfältige Bearbeitung der Fragebögen und besonders auch für die ausführlichen und wertvollen Kommentare. Danke für die freundliche Kooperationsbereitschaft an Dr. Wolfgang Pohl als Geschäftsführer des Bundeswettbewerbs Informatik/Einstieg Informatik, der die anonymisierten Wettbewerbsdaten des Informatik-Bibers 2007 zur Verfügung gestellt hat, und an Eljakim Schrijvers, der den technischen Zugriff auf die Daten ermöglicht hat.

Literaturverzeichnis

- [AK01] Anderson, L.W.; Krathwohl, D.R. (Hrsg.) et al.: A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom's Taxonomy of Educational Objectives. Addison Wesley Longman, New York, 2001.
- [BC82] Biggs, J.B.; Collins, K.F.: Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome). Academic Press, New York, 1982.
- [Bo08] Bower, M.: A Taxonomy of Task Types in Computing. In (ACM Hrsg.): Proceedings of the 13th ITiCSE, Madrid, Spanien, 2008; S. 281-285.
- [Br04] Brinda, T.: Integration of new exercise classes into the Informatics education in object-oriented modelling. In: Education and Information Technologies 9 (2004) 2; S. 117-130.
- [Fr08] Freischlad, S.: Design of Exercises and Test Items for Internetworking Based on a Framework of Exercise Classes. In (Kendall, M.; Samways, B. Hrsg.): Learning to Live in the Knowledge Society. IFIP, Volume 281, Springer, Boston, 2008; S. 261-268.
- [Fu08] Fuller, U. et al.: Developing a Computer Science-specific Learning Taxonomy. In: ACM SIGCSE Bulletin 39 (2007) 4, ACM Press, New York; S. 152-170.
- [GI08] Gesellschaft für Informatik e. V.: Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sek. I. LOGIN 28 (2008) 150/151; Beilage.
- [KF07] Kohl, L.; Fothe, M.: Algorithmen aus einer anderen Perspektive – Ein Vorschlag für ein Kompetenzmodell zum Inhaltsbereich „Algorithmen“ der Bildungsstandards Informatik. LOGIN 27 (2007) 146/147; S. 20-22.
- [Ku04] Kultusministerkonferenz (Hrsg.): Einheitliche Prüfungsanforderungen in der Abiturprüfung Informatik. Luchterhand, München, 2004.
- [La07] Lahtinen, E.: A Categorization of Novice Programmers: A Cluster Analysis Study. In (Sajaniemi, J.; Tukiainen, M. Hrsg.): Proceedings of the 19th annual Workshop of the Psychology of Programming Interest Group, Joensuu, Finnland, 2007; S. 32-41.
- [PF07] Puhmann, H.; Friedrich, S.; Die Standards – und wie weiter? Zur Beurteilung von Aufgaben für Unterricht und Test. LOGIN 27 (2007) 146/147; S. 16-19.
- [Po09] Pohl, W. et al.: Informatik-Biber: Informatik-Einstieg und mehr. Im Tagungsband zur 13. GI-Fachtagung „Informatik und Schule“, INFOS 2009 in Berlin. Bonn, Köllen, 2009.
- [Sc08] Schlüter, K.: Je schwieriger die Aufgabe, desto klüger der Kopf? In (Brinda, T. et al. Hrsg.): Didaktik der Informatik – Aktuelle Forschungsergebnisse, 5. Workshop der GI-Fachgruppe "Didaktik der Informatik", Erlangen 2008. Bonn, Köllen, 2008; S. 77-86.
- [Sh08] Sheard, J. et al.: Going SOLO to Assess Novice Programmers. In (ACM Hrsg.): Proceedings of the 13th ITiCSE, Madrid, Spanien, 2008; S. 209-213.
- [St07] Steinert, M.: Lernzielgraphen und Lernerfolgsanalyse. In (Schubert, S. Hrsg.): Didaktik der Informatik in Theorie und Praxis, INFOS 2007. Bonn, Köllen, 2007; S. 147-158.
- [St08] Starr, C.W. et al.: Bloom's Taxonomy Revisited: Specifying Assessable Learning Objectives in Computer Science. In: ACM SIGCSE Bulletin 40 (2008) 1, ACM Press, New York; S. 261-265.

Was heißt hier Rechnernetz?

Wolf Spalteholz

Didaktik der Informatik
Technische Universität Dresden
01062 Dresden
ws.sideshow@arcor.de

Abstract: Dieser Beitrag diskutiert einen Weg Curricula hinsichtlich des Themengebiets Rechnernetze zu untersuchen. Anschließend wird der sächsische Informatiklehrplan exemplarisch bezüglich dieses Komplexes ausgewertet. Für dieses Vorgehen werden Kriterien zur Auswahl aufgezeigt und beispielhaft Inhalte und Lernbereiche ausgewählt. Mit Hilfe der Lernzielebenen werden exemplarisch Kompetenzen für die Jahrgangsstufen entwickelt, die einen Vergleich zwischen Unterricht, Lehrplan und Informatikstandards möglich machen. Schließlich wird eine Aufgabe aus der bereits vorliegenden Aufgabensammlung am Unterrichtsbeispiel ausgewertet.

1 Einleitung

Naturgemäß ist das Fachgebiet Informatik mit Begrifflichkeiten gespickt, auf deren richtigen Gebrauch man als Informatiklehrer – nicht selten zum Leidwesen der Schüler – Wert zu legen hat. Ein korrekter Umgang mit Fachbegriffen macht es erst möglich, über Informatikinhalte zu sprechen und Neues dazulernen. Geht es jedoch um weiter gefasste informatikbezogene „Schlagworte“, wird es auch bei den Informatikfachkräften oft ungenau. Was sind wichtige Begriffe des Themengebiets Rechnernetze? Fragt man den Administrator eines mittleren Unternehmens, so beinhaltet dieses Thema auf jeden Fall das Gebiet IP-Adressierung und es fallen Begriffe wie „routing“, „MAC-Adresse“ und „portforwarding“. Bittet man eine Informatiklehrkraft einer Universität um einen Kommentar, geht die Antwort eher in Richtung „Topologie“, „Schichtenmodell“ oder „Vermittlungsverfahren“. Werden fachfremde Lehrer gefragt, so steht meist nur die Internetrecherche und das Schreiben und Lesen von E-Mails im Vordergrund. Es muss wohl kaum darauf hingewiesen werden, dass keine der genannten Varianten dem entspricht, was Informatikunterricht im Gebiet Rechnernetze zur Bildung beitragen soll. Doch was genau ist nun im Zusammenhang mit Rechnernetzen Unterrichtsgegenstand? Was sagen Lehrpläne und Standards? Um auf diese Fragen einzugehen wird untersucht, an welchen Stellen dieses Gebiet im (sächsischen) Informatiklehrplan¹ vorkommt, wie

¹Die Lehrpläne Computer und Technik ([Lt06]), Informatik ([Li06]) und die informatikbezogenen Inhalte aus den Profillehrplänen ([Ln06],[Ls06],[Lg06] und [Lk06]) der Klassenstufen 9 und 10 Sachsens werden in diesem Dokument der Einfachheit halber als „sächsischer Informatiklehrplan“ bezeichnet.

es sich in die Inhalts- und Prozessbereiche der Informatikstandards einordnet und wie aus diesen Ansätzen Kompetenzen und schließlich Unterricht entwickelt werden kann. Damit beschreibt dieser Beitrag einen Weg, wie man vom Lehrplan und den Standards zum Unterrichtsinhalt und zu Aufgabenstellungen gelangt.

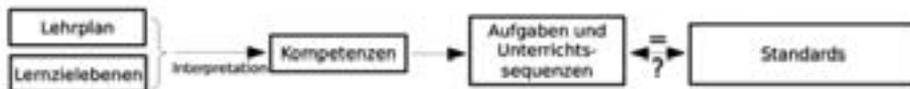


Abbildung 1

Vorgehensweise der Untersuchung

Um eine Schnittmenge aus Standards, Unterrichtspraxis und Lehrplan hinsichtlich des Vorkommens von Rechnernetzinhalt zu untersuchen zu können, muss jeder dieser Bereiche im Hinblick auf das Thema betrachtet und die Ergebnisse auf ein miteinander vergleichbares Niveau gebracht werden. Dabei führt der Weg zuerst zu den aus dem Informatiklehrplan ausgearbeiteten Kompetenzen, die anhand der Lernzielebenen und der Lerninhalte entwickelt werden. Diese sind – wie Abbildung 1 zeigt – in Aufgaben zu überführen und in die Standards einzuordnen. Es soll an dieser Stelle darauf hingewiesen werden, dass mit [Sp08] eine solche Betrachtung in vollständiger Form am Beispiel des sächsischen Lehrplanes vorliegt. Inzwischen wurden einige Aufgabenstellungen in Unterrichtssequenzen erprobt und sollen hier beispielhaft ausgewertet werden.

2 Rechnernetze in den Standards und im Lehrplan

Spätestens seit der Gründung des Vereins „Schulen ans Netz“ beschäftigt sich Schule wenigstens indirekt mit der Thematik der vernetzten Rechnersysteme. Auch die „Alltagstauglichkeit“ dieses Themas wird kaum in Frage gestellt werden; Schülerinnen und Schüler gehen tagtäglich mit solchen Systemen um. Die Rechtfertigung der Beschäftigung mit dem Thema erwächst jedoch nicht direkt aus der Präsenz im Alltag. Vielmehr muss die informatische Grundbildung Kompetenzen im Umgang mit modernen Kommunikationsmitteln sicherstellen. Beispielsweise muss gewährleistet werden, dass die auch von anderen Fächern geforderte Internetrecherche kompetent und kritisch, nicht willkürlich erfolgt. Ferner birgt die Vernetzung enorme Risiken bezüglich des Datenschutzes und der Datensicherheit in sich, denen nur dann in angemessener Weise entgegengetreten werden kann, wenn die zugrunde liegenden Wirkprinzipien der Kommunikation verstanden wurden. Nicht zuletzt ist dieses Thema auch Test- und Übungsfeld anderer informatischer Kompetenzen für Schüler. So wird zum Beispiel durch die Nutzung des Prinzips „Schichtenmodell bei der Kommunikation“ die Kompetenz des Modellierens anschaulich und praxisnah geübt.

Das Thema in den Bildungsstandards

Kompetenzen im Sinne der Bildungsstandards Informatik ergeben sich aus der Verzahnung des Inhaltes (Inhaltsbereiche) mit der Präsentation bzw. der Art des Umgangs mit dem Inhalt (Prozessbereiche). Dieses „*was erschließt sich der Schüler wie*“ erleichtert die Kompetenzformulierung aus den Standards wesentlich.

Das Themengebiet Rechnernetze hat praktisch Einfluss auf jeden Inhaltsbereich. In gleichem Maße lassen sich zu fast jedem Prozessbereich Aufgaben finden, die das Thema tangieren. Im Folgenden soll exemplarisch erläutert werden, wie die Interpretation der Inhalts- und Prozessbereiche hinsichtlich des Leitgedankens Computernetzwerke erfolgte. In der eingangs zitierten Arbeit findet sich sowohl eine Auslegung aller Bereiche hinsichtlich des Themengebiets, als auch einige Beispiele für den Spielraum, die die Interpretation lässt. Es sei darauf hingewiesen, dass Inhalts- und Prozessbereiche nicht getrennt voneinander betrachtet werden können wenn es um Kompetenzentwicklung geht, auch wenn die Interpretation hier in zwei einzelne Abschnitte geteilt wurde. Jeder Inhaltsbereich wird immer *anhand* eines Prozessbereichs interpretiert, was eine wichtige Grundlage für das Erstellen von Aufgaben darstellt.

Inhaltsbereiche

Zur Untersuchung der Inhaltsbereiche wird zu jeder Kompetenz angegeben in welchem Maße sie sich auf das Themengebiet Rechnernetze bezieht. Der Bewertungsmaßstab erstreckt sich von ++ (trifft voll zu) bis -- (trifft gar nicht zu). Die in Tabelle 1 dargestellten Wertungen werden im Anschluss zusätzlich interpretiert. Es handelt sich um den letzten Teilabschnitt des Inhaltsbereichs „Informatiksysteme“ ([St08], S. 17).

Nr.	Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 ...	Wertung
1	...erkennen den Grundaufbau von Informatiksystemen wieder, die in anderen Geräten integriert sind.	0
2	...lösen ähnliche Aufgaben mit unterschiedlichen Programmen der gleichen Anwendungsklasse	++
Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10 ...		
3	...erschließen sich selbstständig neue Anwendungen und Informatiksysteme	+

Tabelle 1

- zu 1) Die informatische Grundbildung in den unteren Jahrgangsstufen beinhaltet die Beschäftigung mit den elementaren Bestandteilen eines PCs. Hier kann ein Hinweis auf die Netzwerkkarte den Bezug zum Themengebiet bilden. Mit diesem Ansatz ist es möglich, rechnernetzbezogene Inhalte in den Unterricht einfließen zu lassen. Grundsätzlich jedoch meint dieser Abschnitt in der

- Hauptsache das Kennenlernen wichtiger Hardwarekomponenten und die Einordnung in das EVA-Prinzip (siehe [St08], S.38).
- zu 2) Hier steht die Benutzung von verschiedenen Browsern, E-Mail-Clients und ggf. auch Datenübertragungs- oder Chat- beziehungsweise Instant-Messaging-Programmen im Vordergrund ([St08], S.39). Da die Beherrschung des Umgangs mit solchen Tools eine grundlegende Fertigkeit der Rechnernetzthematik ist, die als „Handwerkszeug“ später immer wieder benötigt wird, kommt ihnen in den unteren Jahrgangsstufen eine besondere Bedeutung zu.
- zu 3) Im Bereich der Rechnernetze werden im Laufe des Schulalltags diverse Anwendungsprogramme wie die oben Genannten eingesetzt. Eine wichtige Kompetenz im Sinne des selbstständigen Weiterlernens ist, dass Schüler sich solche Anwendungen ohne Hilfe erschließen können. Sie sollen nicht hersteller- oder gar versionsgebundene Handlungsanweisungen auswendig lernen, sondern sich mit den der Anwendung zugrunde liegenden Wirkprinzipien auseinandersetzen. Dies birgt auch die Beschäftigung mit Netzwerkstrukturen und -diensten in sich ([St08], S.40).

Prozessbereiche

Können die Kompetenzen aus den Inhaltsbereichen leicht in tabellarischer Form mit Wertungen versehen werden, fällt die Einschätzung bei den Prozessbereichen wesentlich schwerer: Einerseits ist es hier weit mehr notwendig als bei den Inhaltsbereichen, den ergänzenden Hinweisen in den Standards Aufmerksamkeit zu schenken, da die Prozessbereiche das Wie beschreiben, mit dem sich die Schüler einem Komplex nähern. Andererseits gibt es genauso wenig einen Prozessbereich, der nicht im Bezug auf Rechnernetz Anwendung findet wie einen, der vollständig zum Thema passt. Es soll hier anhand des Prozessbereiches „Begründen und Bewerten“ ([St08], S.48 ff.) in den Jahrgangsstufen 5 bis 7 beschrieben werden, wie eine solche Interpretation aussehen kann. In den unteren Jahrgangsstufen wird das Begründen und Bewerten verwendet, wenn die Schüler die Wahl der Onlinesuchwerkzeuge und –methoden kompetent wiedergeben sollen. Ferner kommt es bei der Beurteilung von Onlineinhalten zum Tragen. Beispielsweise lernen die Schüler Vor- und Nachteile bestimmter Suchwerkzeuge des Internets kennen und können daher zu einem vorgegebenen Inhalt das Werkzeug begründet auswählen. Weiterhin sollen sie in der Lage sein, anhand bestimmter Kriterien die Zuverlässigkeit einer Onlinequelle zu bewerten. Die Beispiele in Tabelle 2 erläutern die Interpretation des Prozessbereiches.

Prozessbereichsformulierung: <i>Die Schüler...</i>	Beispiel
<i>...formulieren Fragen zu einfachen informatischen Sachverhalten</i> <i>...äußern Vermutungen auf der Basis von Alltagsvorstellungen</i>	Die Schüler versuchen, anhand ihres Alltags- und Vorwissens zu bestimmten Suchbegriffen das richtige Suchwerkzeug auszuwählen.

Prozessbereichsformulierung: <i>Die Schüler...</i>	Beispiel
<i>...nennen Vor- und Nachteile</i> <i>...können Argumente nachvollziehen</i> <i>...bewerten Informationdarstellungen hinsichtlich ihrer Eignung</i>	Die Schüler vergleichen die Suchstrategien „Suche in einem Lexikon“ und „Internetrecherche“ miteinander. Sie stellen Unterschiede und Gemeinsamkeiten sowie Vor- und Nachteile beider Möglichkeiten heraus.

Tabelle 2

Auswahlkriterien für Lernbereiche und Lerninhalte des Lehrplanes

Um Lehrplaninhalte in Kompetenzen zu überführen, über die die Schüler nach jeder Jahrgangsstufe verfügen sollen, werden Kriterien zu deren Auswahl benötigt. Es sei bereits an dieser Stelle darauf hingewiesen, dass sich diese Kompetenzen nicht direkt mit den in den Standards formulierten vergleichen lassen, da „die vorliegenden Informatikstandards Mindeststandards [sind]“ ([St08], Seite V). Bei den Lernzielen und Lerninhalten im Lehrplan handelt es sich jedoch im Allgemeinen um Regelanforderungen. Außerdem werden die in den Inhalts- und Prozessbereichen beschriebenen Anforderungen in die Jahrgangsstufen 5-7 und 8-10 geteilt, die Lehrpläne differenzieren jedoch für gewöhnlich in einzelne Schuljahre. Ferner reicht der Lehrplan bis zur Klassenstufe 12 (bzw. 13). Der Vergleich lässt sich also nur indirekt ziehen.

Es ließe sich vorzüglich eine Kontroverse darüber anstellen, ob man sich mit Rechnernetzen beschäftigt, wenn man ein Chatprogramm nutzt. Ebenso ist es Dank moderner Betriebssysteme möglich, Rechner miteinander zu verbinden, ohne auch nur über Grundlagenkenntnisse auf diesem Gebiet zu verfügen. Um einen Lehrplan hinsichtlich eines Themenkomplexes untersuchen zu können, müssen daher Kriterien vorhanden sein, mit denen für einen Lernbereich entschieden werden kann, ob er zur Interpretation in Frage kommt oder nicht. In dieser Betrachtung soll Folgendes gelten:

Umfang: Die fachlichen Inhalte des Lernbereichs weisen einen zeitlichen Bezug zum Thema Rechnernetze zu wenigstens 1/3 auf. Hier wurden, um einen Prozentwert zu ermitteln, in der Tat die Lerninhalte und Lernziele gezählt um die vorgesehene Stundenzahl abschätzen zu können.

Fachlichkeit: Die Kompetenzen, die die Schüler erreichen sollen, stehen im direkten Bezug zum Themengebiet Rechnernetze und haben sowohl praktischen als auch theoretischen Bezug zum Gebiet. Das heißt, die Lernbereichsbeschreibung umfasst unmissverständlich ein Auseinandersetzen mit Rechnernetzinhalten.

Pflichtkanon: Schließlich wurden nur Inhalte ausgewählt, mit denen sich alle Schüler auseinandersetzen. Diese Regelung betrifft beispielsweise die Klassenstufen 9 und 10 in Sachsen, in denen die Gymnasiasten in den Profilen (naturwissenschaftliches, gesellschaftswissenschaftliches, künstlerisches oder sportliches) unterrichtet werden.

Ausgewählte Inhalte am Beispiel des sächsischen Lehrplanes

Es soll hier stellvertretend der sächsische Informatiklehrplan hinsichtlich zu realisierender Themen im Komplex Rechnernetze untersucht werden, da momentan nur dort eine Untersuchung über alle Schuljahre hinweg möglich ist. Tabelle 3 zeigt die Lernbereiche, die den oben genannten Kriterien entsprechen. Es sei dringend darauf hingewiesen, dass die umgangssprachliche Formulierung des Inhaltes schon daher ungenau sein muss, weil sie in einem Satz den Inhalt mehrerer Unterrichtsstunden angibt.

Klasse	Lernbereich	Umgangssprachliche Inhaltsformulierung
5	Informationsbeschaffung mit dem Computer ca. 3 von 12 Ustd.	Browser kennen und bedienen, WWW-Dienst des Internets kennen lernen
6	Informationsaustausch mit dem Computer ca. 3 von 6 Ustd.	E-Mail-Verkehr beherrschen, erste Hinweise auf Sicherheitsbedarf und Notwendigkeit der Inhaltskritik.
7	Computer benutzen – Elemente und Strategien ca. 4-5 von 14 Ustd.	Internetrecherche und Umgang mit Online-suchwerkzeugen, Qualitätsbeurteilung und Qualitätsmerkmale von Suchergebnissen
8	Daten verarbeiten ca. 2-3 von 8 Ustd.	Erste theoretische Auseinandersetzung, Client-Server-Prinzip
9/10	je nach Profil	Topologien, praktischer Netzaufbau, einfaches Schichtenmodell, sonst: je nach Profil
11/12	Kommunikation in Netzen 8 Ustd.	tiefere Beschäftigung mit Theorie, Protokolle, (OSI)-Schichtmodell, IP-Adressierung; Weitere Auseinandersetzung mit Rechnernetzanwendungen (ftp-Client etc.)

Tabelle 3

Ermittlung informatischer Kompetenzen:

Die Lernzieloperationalisierung – also die Entwicklung von zu erreichenden Kompetenzen aus dem Lehrplan – entsteht aus der Verbindung „Lerninhalt – Lernzielebene“. Es geht an dieser Stelle um eine Zielformulierung, die es dem Lehrer einerseits ermöglicht Unterricht zu gestalten und andererseits, sofern die Unterrichtsziele erreicht werden, das Wissens- bzw. Kompetenzniveau des Schülers so genau wie möglich zu beschreiben. Die aus dem Lehrplan extrahierten Inhalte können natürlich nicht ohne den Blick auf den Unterricht in Kompetenzen überführt werden, denn Vorwissen, Motivation und Leistungsstärke der Schüler sind nur einige äußere Faktoren, die diese Aufgabe beeinflussen. Die Kompetenzen sind zwar nicht auf einem „Stundenziel“-Niveau dargestellt, allerdings sind sie detaillierter als die Lehrplanformulierungen. Hat man eine Aufgabe entwickelt, so stellt die Kompetenz *die minimale Differenz zweier Schüler dar, von denen einer die Aufgabe lösen kann und der andere nicht*. Abbildung 2 zeigt beispielhaft die entwickelten Kompetenzen zum Lehrplan der Klassenstufe 6 in Sachsen. Im Lernbereich 2 – *Informationsaustausch mit dem Computer* – formuliert der Lehrplan: „*Beherrschen einer Form der elektronischen Kommunikation zum gemeinsamen Arbeiten*“.

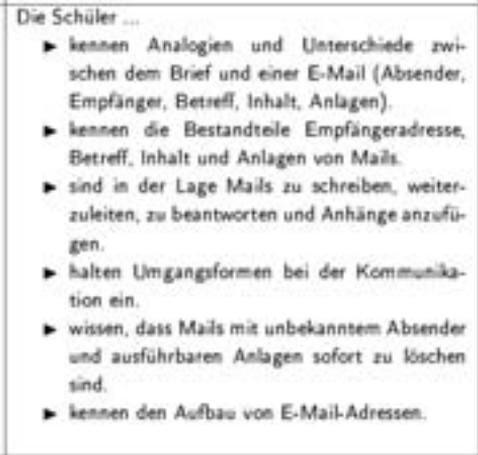
- 
- Die Schüler ...
- ▶ kennen Analogien und Unterschiede zwischen dem Brief und einer E-Mail (Absender, Empfänger, Betreff, Inhalt, Anlagen).
 - ▶ kennen die Bestandteile Empfängeradresse, Betreff, Inhalt und Anlagen von Mails.
 - ▶ sind in der Lage Mails zu schreiben, weiterzuleiten, zu beantworten und Anhänge anzufügen.
 - ▶ halten Umgangsformen bei der Kommunikation ein.
 - ▶ wissen, dass Mails mit unbekanntem Absender und ausführbaren Anlagen sofort zu löschen sind.
 - ▶ kennen den Aufbau von E-Mail-Adressen.

Abbildung 2

3 Aufgabensammlung und Unterricht

Es fällt seit der Verabschiedung der „Grundsätze und Standards für die Informatik in der Schule“ [St08] ein anderes Licht auf viele Informatikfachgebiete. Nun sind nicht mehr nur die Lehrpläne wegweisend für Inhalt und Tiefe des Unterrichtsstoffes: Mit den Standards wird (leider unverbindlich) festgelegt, welche Informatikkompetenzen jeder Schüler am Ende bestimmter Jahrgangsstufen aufweisen soll und dies gilt selbstverständlich auch für das Wissen und den Umgang mit Rechnernetzsystemen.

Das zeigt, dass Auseinandersetzungen mit dem Themengebiet und dessen Umsetzung im Unterricht durchaus aktuell sind. Die Sammlung in [Sp08], die 41 Aufgaben mit Inhalten zu Rechnernetzen enthält, dient als Grundlage, Vergleiche mit den Informatikstandards anzustellen. Die Aufgaben wurden, wie in Abbildung 3 zu sehen ist, systematisch nach Kompetenzen klassifiziert. Zwar steht ein Test der Aufgaben und daraus entwickelter Unterrichtssequenzen in weiten Teilen noch aus, es wurde jedoch bereits begonnen gemeinsam mit Lehrern die Aufgaben im Unterricht zu erproben. Ausgangspunkt für die Entwicklung der Aufgaben waren jeweils die aus dem Lehrplan entwickelten Kompetenzen. Das Ziel bestand darin, diese so genau wie möglich zu prüfen bzw. auf deren Erwerb hinzuwirken.

09_RN_03	
Aufgabenthema	Hardware in Rechnernetzen
Kompetenzen	Die Schüler kennen und erfassen Unterschiede zwischen Hubs und Switches.
Klassenstufe	9/10
Lernzielebene	Übertragen / Anpassen
Zeitrhythmus	10 min

Abbildung 3

4 Beispielaufgabe

Aufgabenstellung

Netzwerkknoten

Auf dem Boden einer Firma findet ein Mitarbeiter zwei Netzwerkverteiler. Leider steht auf keinem der beiden, ob es sich um einen Hub oder einen Switch handelt. Der Mitarbeiter beschließt daher, die beiden Geräte in verschiedenen Situationen zu testen.

Situation 1: Der Mitarbeiter schließt zwei Rechner an und kopiert eine größere Datenmenge von einem Rechner zum anderen.

Situation 2: Der Mitarbeiter schließt vier Rechner an und kopiert dieselbe Datenmenge gleichzeitig vom ersten zum zweiten und vom dritten zum vierten Rechner.

Die Tabelle unten zeigt die Dauer des Kopiervorgangs für beide Situationen mit beiden Geräten. Entscheide, bei welchem Gerät es sich um einen Switch und bei welchem es sich um einen Hub handelt und begründe Deine Antwort.

	Gerät A	Gerät B
Situation 1	124s	136s
Situation 2	231s	141s



Erwartungsbild

Bei Gerät A handelt es sich um einen Hub. Gerät B ist ein Switch.

Gerät B benötigt in Situation 1 mehr Zeit als Gerät A, da die Datenpakete ausgelesen werden, um den Empfänger zu bestimmen und sie zu vermitteln. Gerät A macht das nicht und benötigt daher weniger Zeit. In Situation 2 benötigt Gerät A mehr Zeit als Gerät B, da die zu übertragenden Daten jeden Rechner erreichen und daher nicht gleichzeitig vom ersten zum zweiten und vom dritten zum vierten Rechner übertragen werden können. Bei Gerät B werden die Daten parallel von Rechner eins zu Rechner zwei und von Rechner drei zu Rechner vier übertragen. Daher benötigt es weniger Zeit für die gesamte Übertragung.

Unterrichts- und Vorbedingungen

Die Aufgabe wurde an einem sächsischen Gymnasium in der Klassenstufe 10 eingesetzt.

Im Rahmen des Unterrichts, der in diesen Klassenstufen integrativ in Profilen stattfindet, sind insgesamt etwa 10 Stunden für den Rechnernetz-Komplex vorgesehen. An der Erprobung nahm eine Klasse mit 15 Schülern (sechs Mädchen, neun Jungen) teil.

Vor der Bearbeitung der Aufgabe behandelten die Schüler neben den wesentlichen Netzwerktopologien sowie deren Vor- und Nachteile die Netzarten WAN und LAN, die grundlegenden Netzwerkknoten (Repeater, Switch, Hub, Router) sowie deren Aufgaben und definierten den Begriff „Computernetz“. Ferner wurden die für das Lösen der Aufgabe wichtigsten Merkmale – das Weiterleiten der Datenpakete an alle Rechner (Hub) bzw. nur an die Zieladresse (Switch) – erläutert.

Die Schüler kannten folglich den Funktionsunterschied zwischen Hub und Switch (Aufgabe und Wirkungsweise). Der Zeitfaktor beim Übertragen von Daten wurde vom Lehrer jedoch nicht thematisiert. Aus dem Unterrichtsgespräch war abzuleiten, dass den Schülern die Tatsache bekannt war, dass Daten in Computernetzwerken in Paketen übertragen werden. Die Leistung der Schüler bestand also aus dem Schlussfolgern vom Rechenaufwand der Adressierung hin zum Mehraufwand an Zeit beim Switch in Situation A sowie der kürzeren Übertragungszeit in Situation B aufgrund der Möglichkeit mehrere Übertragungen gleichzeitig laufen zu lassen.

Auswertung der Aufgabe

Teil A (Gerätezuordnung):	86% richtige Antworten
Teil B (Begründung):	57% als richtig gewertete Antworten

Anzahl	Fehler
2	Keine Begründung abgegeben (Teil A war jeweils richtig gelöst).
2	Falsch argumentiert, Teil A war ebenfalls falsch.
1	Zeichnung abgegeben, die jedoch keine Begründung enthielt.
2	Unvollständig argumentiert.

Tabelle 4

Die überwiegende Mehrheit der Schüler konnte beim Test die Geräte richtig zuordnen. Zur Prüfung der Kompetenz „...kennen und erfassen Unterschiede zwischen Hubs und Switches“ gehört es auch, dass die Schüler diese Unterschiede benennen. Die Auswertung der Fehler in Aufgabenteil B zeigt, dass es neben fachlichen Unsicherheiten auch Probleme mit der Kommunikation bezüglich informatischer Inhalte bzw. bei der begründeten Stellungnahme im Allgemeinen gab (siehe Tabelle 4). Abbildung 4 stellt eine als falsch gewertete Antwort eines Schülers dar. Hier war die Zuordnung Hub → Gerät A und Switch → Gerät B richtig, jedoch konnte der Schüler seine Behauptung nicht in der geforderten Art und Weise begründen.

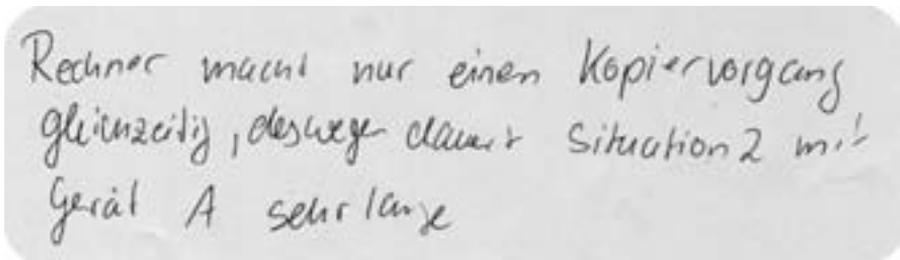


Abbildung 4

Dies macht die Wichtigkeit der beiden Teile der Aufgabe deutlich: Teil A prüft nicht die vollständige Kompetenz „...kennen und erfassen Unterschiede zwischen Hubs und Switches“, gibt allerdings einen Anhaltspunkt darauf, ob der Schüler in der Lage ist, richtig zuzuordnen oder nicht. Teil B, der diese Kompetenz im ganzen Umfang verlangt, fordert vom Schüler aber das Begründen und das Kommunizieren über informatische Inhalte. Es werden also die Prozessbereiche Begründen und Bewerten sowie Kommunizieren und Kooperieren tangiert.

5 Fazit und Ausblick

Zielstellung der Beschäftigung mit dem Thema Rechnernetze war es, darzulegen an welchen Stellen der Komplex im Lehrplan auftritt und wie man daraus Unterricht entwickeln kann. Der Weg dieser Entwicklung führte zu einer Zahl an Kompetenzen, die die Schüler im Laufe ihrer Ausbildung erreichen sollen. Vorteil dieses Entwicklungsweges ist es, dass so aus dem Lehrplan Kompetenzen entstehen, deren Erreichen einerseits mit den entwickelten Aufgaben valide überprüft werden kann und die andererseits den Vergleich mit den Informatikstandards (und somit das Ausrichten an Mindeststandards) möglich machen. Dieses Vorgehen wurde an einem Themenbereich gezeigt und es liegt mit [SP08] – bezogen auf den Lehrplan des Landes Sachsen – eine vollständige Interpretation einschließlich Aufgabensammlung über alle Jahrgangsstufen vor. Das beschriebene Vorgehen kann, wie beispielsweise [Wi08] zeigt, auch auf andere Informatikinhalte übertragen werden.

Der nächste Schritt dieser Entwicklung ist nun, die Aufgaben im Unterrichtsalltag zu testen und gegebenenfalls anzupassen. Zu klärende Fragen sind, ob die Aufgaben die Kompetenzen wirklich prüfen und ob die Schüler mit Hilfe des auf den Aufgaben aufbauenden Unterrichts diese auch erwerben. Nur mit Hilfe empirischer Tests kann ein Kompetenzmodell für das Themengebiet Rechnernetze entstehen das fixiert, welche Fähigkeiten und Fertigkeiten die Schüler über alle Jahrgangsstufen zu erreichen haben. Ferner ist eine Unterscheidung zwischen Aufgaben für den Unterricht und Aufgaben in Testsituationen vorzunehmen, um einerseits Erwerb und andererseits Überprüfung der Kompetenzen zu ermöglichen. Erste Proben haben gezeigt, dass die Aufgaben durchaus unterrichtsrelevant sind: Sie lassen sich in den Unterricht einfügen und helfen beim Erreichen der gesetzten Ziele.

Ebenfalls steht ein vollständiger Vergleich dieser entwickelten Kompetenzen mit den Informatikstandards noch aus. Einige Aufgaben wurden exemplarisch zu Inhalts- und Prozessbereichen zugeordnet, jedoch sind, um einen vollständigen Vergleich sinnvoll zu bewerkstelligen, erst der Praxistest der Aufgaben und eventuelle Änderungen abzuwarten.

Literaturverzeichnis

- [Lg06] Comenius-Institut: Profillehrplan gesellschaftswissenschaftliches Profil – Gymnasium. Internetveröffentlichung.
http://www.sn.schule.de/ci~/download/lp_gy_gesellschaftswissenschaftliches_profil.pdf
Revision: 13.01.2006
- [Li06] Comenius-Institut: Lehrplan Informatik – Gymnasium. Internetveröffentlichung.
http://www.sn.schule.de/ci~/download/lp_gy_informatik.pdf
Revision: 13.01.2006
- [Lk06] Comenius-Institut: Profillehrplan künstlerisches Profil – Gymnasium. Internetveröffentlichung.
http://www.sn.schule.de/ci~/download/lp_gy_kuenstlerisches_profil.pdf
Revision: 13.01.2006

- [Ln06] Comenius-Institut: Profilehrplan naturwissenschaftliches Profil – Gymnasium. Internetveröffentlichung.
http://www.sn.schule.de/ci~/download/lp_gy_naturwissenschaftliches_profil.pdf
 Revision: 13.01.2006
- [Ls06] Comenius-Institut: Profilehrplan sportliches Profil – Gymnasium. Internetveröffentlichung.
http://www.sn.schule.de/ci~/download/lp_gy_sportliches_profil.pdf
 Revision: 13.01.2006
- [Lt06] Comenius-Institut: Lehrplan Technik/Computer – Gymnasium. Internetveröffentlichung.
http://www.sn.schule.de/ci~/download/lp_gy_technik_computer.pdf
 Revision: 13.01.2006
- [Sp08] Spalteholz, W.; Standardisierung in der informatischen Bildung am Beispiel des Themengebiets Rechnernetze, Wissenschaftliche Staatsexamensarbeit, Dresden 2008
 Internetveröffentlichung: <http://sideshow.homelinux.net/inhalt/sea.pdf>
- [St08] H. Puhlmann, S. Friedrich et al. Arbeitskreis Bildungsstandards des Fachausschusses informatische Bildung in Schulen (FA IBS) und der FG Didaktik der Informatik (FG DDI): Grundsätze und Standards für die Informatik in der Schule – Bildungsstandards Informatik (Stand: 24. Januar 2008).
- [Wi08] Windisch, M.; Untersuchung zur informatischen Bildung an Schulen Sachsens – die Bildungsstandards und der Lehrplan: Vergleich, Interpretation, Kopplung und Umsetzbarkeit, Wissenschaftliche Staatsexamensarbeit, Dresden 2008

Zurück in die Zukunft?

Zur Geschichte der Rahmen(lehr)pläne Informatik SekII in Berlin (West)

Lothar Sack
Dickhardtstr. 26
12161 Berlin
LotharSack@t-online.de

Helmut Witten
Brandenburgische Str. 23
10707 Berlin
helmut@witten-berlin.de

Abstract: Am 8.-10. Oktober 1984 fand an der FU Berlin die erste GI-Fachtagung zur Informatik in der Schule statt [AH84]. Nach 25 Jahren kehrt diese Tagung, für die sich inzwischen die Abkürzung INFOS eingebürgert hat, wieder nach Berlin zurück, das Motto lautet *Zukunft braucht Herkunft - 25 Jahre INFOS*. Dieser Leit-spruch soll mit dem vorliegenden Beitrag an einem konkreten Beispiel aufge-griffen werden: der Geschichte der Rahmen(lehr)pläne Informatik SekII in Berlin (West). Beide Autoren haben über viele Jahre an der Entwicklung der Berliner Rahmenpläne mitgewirkt. Das ermöglicht einen exemplarischen Längsschnitt zur Entwicklung der Rahmenpläne in einem Bundesland.

1 Ein Blick 25 Jahre zurück

Auf der GI-Tagung 1984 hielten Wilfried Koch und Lothar Sack ein Referat über Konzepte und Inhalte für einen Lehrplan Informatik in der gymnasialen Oberstufe [KS84]. Dieser Rahmenplan wurde zum Schuljahr 1984/85 in Kraft gesetzt und galt fast 10 Jahre [S85], bis er durch den Nachfolgeplan [S93] ersetzt wurde. Die ersten Gehversuche zum Informatikunterricht in Berlin lagen 1984 bereits 15 Jahre zurück, in den Jahren 1972 [S72] und 1977 [S77] waren Vorgänger-Rahmenpläne erschienen, wobei der 77er Plan auf der Grundlage eines Modellversuches an der Fritz-Karsen-Schule in Berlin-Neukölln erarbeitet worden war [S78]. Während im Plan von 1972 lediglich zwei Kurshalbjahre beschrieben wurden, waren es 1977 bereits vier Semester. Im Plan von 1985, der uns im ersten Teil dieses Artikels schwerpunktmäßig beschäftigen wird, sind drei Lernjahre Informatik für die Klassenstufen 11-13 beschrieben. Es finden sich nur Grundkurse in diesem Plan, Leistungskurse waren den späteren curricularen Entwicklungen vorbehalten.

Erfahrungen mit Unterricht und Schülerverhalten bei den frühen Rahmenplänen führten zu grundsätzlichen Überlegungen, die in den Rahmenplänen von 1984 [S85] und 1993 [S93] berücksichtigt wurden: Jeweils ein Drittel der Schüler setzte den Unterricht im

folgenden Unterrichtsjahr nicht fort. Ursache hierfür war nicht etwa ein schwindendes Interesse an der Informatik, sondern die geringe Wertigkeit, die dem Fach Informatik im Fächerverbund des dritten Aufgabenfeldes zugebilligt wurde. Abgesehen von der Wahl als Abiturprüfungsfach war der Besuch von mehr als zwei Informatik-Kursen für viele Schüler unter abiturtaktischen Gesichtspunkten nur interessant, wenn sie überdurchschnittliche Noten erhielten. So konkurrierte das Fach Informatik etwa mit dem Fach Sport.

Das führte zu zwei Überlegungen: Erstens erschien es konsequent, nach jedem Unterrichtsjahr eine sinnvolle inhaltliche Abrundung anzustreben, so dass die Schüler auch bei nicht vollständigem Durchlauf durch die Kursfolge mit einem (einigermaßen) zutreffenden Bild von der Informatik und ihren Bezügen zur realen Welt diesen Unterricht verließen. Zweitens erschien es sinnvoll, die Schüler in unterschiedlichen Rollen zur Informatik zu sehen: als von Informatik-Anwendungen Betroffener, als Benutzer von Informatik-Systemen, als jemand, der Informatiksysteme analysiert und versteht, sowie als Konstrukteur solcher Systeme. In jedem Kurs sollten die verschiedenen Rollen angesprochen und bewusst gemacht werden, mit fortschreitendem Unterricht trat die Rolle des kompetenten Konstrukteurs stärker in den Vordergrund.

1.1 Anwendungen und Auswirkungen

Das Konzept des anwendungsorientierten Informatikunterrichts wurde in Berlin in den 70er-Jahren entwickelt (vgl. [GR76], [SZ78]):

In der Entwicklung der Curriculum-Reform wurde die traditionelle Konzeption des durch die jeweilige Fachsystematik strukturierten Unterrichts als nicht mehr vereinbar mit der Erziehungs- und Ausbildungsfunktion der allgemeinbildenden Schule erklärt. Die Zielsetzung, Erziehung als Ausstattung zur Bewältigung von Lebenssituationen zu begreifen, zog die Forderung nach situationsbezogener Curriculumentwicklung nach sich [...].

Für den Unterricht in den Naturwissenschaften, der Mathematik und der Informatik wurde an mehreren Stellen diese Diskussion aufgegriffen und ein gesellschaftsbezogener, an den Lebenssituationen der Schüler orientierter Unterricht konzipiert [...].

Der anwendungsorientierte Informatikunterricht fußt auf dieser Konzeption. Intendiert wird nicht die Herausbildung einer allgemeinen Methode zur Lösung von Problemen beliebigen Inhalts. Vielmehr ist der durch den Einsatz der Datenverarbeitung in unserer Gesellschaft betroffene Schüler zu berücksichtigen und zur Selbst- und Mitbestimmung zu befähigen ([SZ78], S. 43).

In dem Beitrag von Koch/Sack zur Infos'84 heißt es entsprechend:

Daneben müssen Schüler einer allgemeinbildenden Schule über heutige und zukünftige Anwendungen der Informationstechnologie in fast allen gesellschaftlichen Bereichen informiert sein; dies umso mehr als eines der wesentlichen Motive zur Einführung dieses Fachs gerade die gesellschaftliche Virulenz der Rechneranwendungen war und ist. Diese Information schließt die Kenntnis der Veränderungen mit ein, welche der Einsatz der Informationstechnologie bewirkt hat und bewirken wird. Schließlich sollen die Möglichkeiten

und Grenzen der Einflussnahme und Gestaltung beim Rechneinsatz für Betroffene und aktiv Beteiligte aufgezeigt werden ([KS84], S. 135).

Im Rahmenplan [S85] werden diese Themen als *Anwendungen und Auswirkungen* bezeichnet, der dafür ausgewiesene Stundenanteil beträgt ca. 25%. Im ersten Unterrichtsjahr sollen die Lernenden einen fundierten Überblick über typische Anwendungen der Informatik erhalten, Auswirkungen auf bestehende Berufe und das Thema Datenschutz sollen behandelt werden ([S85], S. 11). Im zweiten Lernjahr soll dies vertieft und durch Informationen über typische Datenverarbeitungsberufe ergänzt werden ([S85], S. 26).

Der anwendungsorientierte Ansatz wurde in der Fachdidaktik schon lange kontrovers diskutiert. So schrieb Rüdiger Baumann in [B96]:

Die Lehrer naturwissenschaftlicher Fächer sind von ihrer Sozialisation, ihren Interessen und ihrer Fachausbildung her gar nicht in der Lage, historische oder ethisch-politische Themen im Unterricht zu behandeln. [...] Angesichts dieser Sachlage stellt sich die Frage, wieso die genannte Aufgabe nicht von den hierfür zuständigen Fächern wahrgenommen wird (S. 134).

Bei dieser pessimistischen Sicht der Dinge stellt sich allerdings die Frage, ob fachübergreifendes und fächerverbindendes Lernen, wie es auch von der KMK gefordert wird, mit dieser Haltung wirksam möglich ist. Die Wirklichkeit ist nicht nach Fächern sortiert. Wenn Fachlehrerinnen und -lehrer darauf beharren, nur ihre Fachsicht zu vermitteln, werden die Lernenden mit der in den Präambeln der Lehrpläne geforderten Bewältigung der schwierigsten Aufgabe allein gelassen, nämlich die verschiedenen Perspektiven zu integrieren und auf praktische Lebenssituationen zu beziehen. Darüber hinaus kann man mit guten Gründen bezweifeln, ob sich z. B. die Sozialkunde dieser Aufgabe schon aus zeitlichen Gründen überhaupt widmen kann. Auch müsste man dem Sozialkundelehrer in analoger Weise zugestehen, sich auf seine fehlende Sozialisation und Kompetenz in Fragen der Informatik zurückzuziehen.

1.2 Algorithmik

Die Algorithmik wurde vor 25 Jahren als „Hauptgegenstand und -methode der Informatik“ gesehen ([S85], S. 8). Damit stellte sich der Berliner Rahmenplan in die Tradition des algorithmenorientierten Unterrichts, der nicht zuletzt durch die GI-Empfehlungen *Zielsetzungen und Inhalte des Informatikunterrichts* aus dem Jahr 1976 zum *Mainstream* innerhalb der Bundesrepublik geworden war. Diese Gewichtung spiegelte sich auch im vorgesehenen Stundenvolumen für diese Thematik, mehr als 50% der Unterrichtszeit waren dafür vorgesehen.

Im ersten Unterrichtsjahr war die „Einführung in grundlegende Elemente der Algorithmisierungsmethodik an Hand kleiner (sog. stand alone-) Programme“ vorgesehen ([S85], S. 11). Dabei sollte in die Methode der strukturierten Programmierung eingeführt werden, ein Programmier(sprachen)kurs wurde abgelehnt. „In jedem Fall sollen Aufgaben aus den Anwendungsbereichen gewählt werden, die für die Informatikanwendung reprä-

sentativ sind. [...] Mathematische und physikalische Problemstellungen sind nur sparsam zu benutzen“ (ebd., S. 12).

Im zweiten Unterrichtsjahr stand einerseits der Prozedur- und Modulbegriff im Vordergrund, es sollten damit Methoden erlernt werden, „die geeignet sind, komplexere algorithmische Aufgaben zu bearbeiten“ ([S85], S. 26). Zum anderen sollte „der Schüler [...] eine Reihe von Standardalgorithmen kennen lernen, die in vielfältigen Anwendungen eine Rolle spielen“ (ebd.). Ein wichtiger Aspekt waren die Spezifizierung und die Validierung der Programme, Prozeduren und Modulen ([KS84], S. 136). Darüber hinaus sollte zur Vorbereitung des Projektunterrichts auch in den *Software-Life-Cycle* eingeführt werden ([S85], S. 33).

Problematisch war und blieb die Einführung in die Algorithmik anhand von sog. *Micky-Maus-Programmen*. Hier wurde mit dem Folgeplan [S93] Neuland betreten. Als Einstieg war nunmehr die Benutzung und Analyse eines dokumentierten Systems mittlerer Komplexität vorgesehen: „Durch diesen Einstieg wird schon frühzeitig das Denken in komplexen Zusammenhängen gefördert“ ([S93], S. 12). Dieser Ansatz wurde später von der Paderborner Schule der Informatik-Didaktik mit der *Dekonstruktion von Informatiksystemen als Unterrichtsmethode* weiter ausgebaut und in Richtung Objektorientierung vertieft [HMS99].

1.3 Rechnerorganisation und Rechnerbedienung

Der erste Informatik-Rahmenplan in Berlin, das *vorläufige Grundprogramm* [S72], ist typisch für den rechnerorientierten Ansatz (vgl. [GR76], [SZ78]). Entsprechend der seinerzeit zur Verfügung stehenden Rechentechnik stand die maschinennahe Programmierung im Vordergrund, angestrebtes Bildungsziel war die *Entmystifizierung des Elektronengehirns*. Bei Helmar Frank, einem früher Verfechter dieses Ansatzes, hieß die Informatik bezeichnenderweise noch „Rechnerkunde“. Er forderte die Kenntnis einer maschinennahen Sprache als notwendige Voraussetzung zum Erlernen einer problemorientierte Sprache ([SZ78], S. 41f.).

Der rechnerorientierte Ansatz wurde schon im folgenden Rahmenplan [S77] relativiert. Vor 25 Jahren wurde der Rechner in erster Linie als Werkzeug für den Unterricht gesehen:

Die Beschäftigung mit den Komponenten eines Rechnersystems muss sich jedoch stets den anwendungsbezogenen und algorithmischen Zielsetzungen des Unterrichts unterordnen. Leitfaden für den Unterricht sollte das im Umgang mit dem Rechner wachsende Bedürfnis des Schülers sein, auch Einblick in die Funktionsweise von Rechnersystemen zu erhalten ([S85, S. 9]).

In der Rechnerorganisation besteht die Gefahr, dass die Eigenschaften des konkreten Schulrechners zum Leitfaden des Unterrichts werden. Aus diesem Grund wird im neu erstellten Rahmenplan zwischen Rechnerorganisation als Unterrichtsgegenstand und den notwendigen Kenntnissen der Rechnerbedienung, deren Erwerb in den Unterricht zu integrieren ist, strikt unterschieden ([KS84], S.135).

Im Rahmenplan [S85] werden daher für die Rechnerbedienung keine eigenen Unterrichtszeiten ausgewiesen, für die Rechnerorganisation waren ca. 22% der zur Verfügung stehenden Zeit vorgesehen. Dieser Prozentsatz wurde im Folgeplan [S93] mit ca. 9% nochmals stark reduziert, eine ausführliche Behandlung der technischen Informatik war nur als Vertiefungsgebiet möglich.

1.4 Software-Engineering und Projektunterricht

Das wichtigste und in dieser Konsequenz neue Paradigma für den vor 25 Jahren vorgestellten Rahmenplan war die Verbindung von Projektunterricht mit Ideen aus dem seinerzeit noch jungen Fachgebiet Software Engineering (vgl. [JKS83]). Zur Frage, warum Projekte im Informatikunterricht durchgeführt werden sollten, schrieben Stefan Jänichen (heute Präsident der GI), Wilfried Koch und Gerd Schürmann aufgrund ihrer Erfahrungen in der Lehre an der TU Berlin:

Ohne Projekte ist die reale Beschränktheit der Ressourcen (z. B. Zeit, Qualität der Werkzeuge, Kenntnisstand der Beteiligten) bei der Herstellung eines Produktes nicht erkennbar. Auch noch so gute Planspiele können diese Erfahrung nicht ersetzen. [...] Durch ein Projekt können die unterschiedlichen Interessen vom Auftraggeber für ein Produkt, Hersteller eines Produktes, des Benutzers und des von einem Produkt Betroffenen deutlich gemacht werden. [...]

Weitere wichtige Erfahrungen werden Lehrer und Schüler aus einem guten Projektunterricht gewinnen. Hier ist besonders der Wert organisierter Teamarbeit zu nennen, aber auch fächerübergreifende Integrationsmöglichkeiten des Projektunterrichts u. ä.

Wir sind der Meinung, dass in keinem der klassischen Schulfächer die Chance so groß ist, dieses „Neuland“ entschlossen zu betreten, wie im Informatikunterricht. So ist heute im Informatikunterricht noch genügend Spielraum, Problemstellungen anderer Fächer in einem Projekt anzugeben. Werden wir die Chance nutzen oder aber die Informatik ebenfalls in das enge Korsett der Lehrpläne zwingen, aus dem die anderen Schulfächer nicht mehr herauskommen? ([JKS83], S. 26)

Wie wir weiter unten sehen werden, ist in den letzten 25 Jahren auch für die Informatik in Berlin das Korsett enger geworden. Doch zurück ins Jahr 1984: Welches Verständnis von Software-Engineering lag dem damaligen Rahmenplan zu Grunde? Es handelte sich um ein didaktisch vereinfachtes Wasserfallmodell:

Gegenstand des Unterrichts ist die phasenbezogene Vorgehensweise bei der Softwareerstellung:

- Problemanalyse,
- funktionelle Analyse,
- Entwurfsanalyse (Modularisierung),
- Modulprogrammierung und Modultest,
- Systemintegration,
- Beurteilung und Einsatz des Systems.

Jeder Arbeitsschritt zielt auf die Herstellung eines abschließenden Dokuments, das Ausgangspunkt für den nächsten Arbeitsschritt ist:

- Anforderungsdefinition als abschließendes Dokument der Problemanalyse,
- funktionelle Spezifikation als Ergebnis der funktionellen Analyse,
- Entwurfspezifikation als Ergebnis der Entwurfsanalyse,
- Modulprogramme und Testprotokolle als Ergebnis von Modulprogrammierung und -test,
- einsatzfähiges und dokumentiertes System als Ergebnis der Systemintegration,
- dokumentiertes System mit gezielten Bewertungen und Änderungsvorschlägen als Ergebnis der Beurteilung und des Einsatzes des Systems ([KS84], S. 138).

Das Wasserfallmodell ist heutzutage sicher nicht mehr state of the art, die Nachteile dieses Vorgehensmodells sind in der Praxis vielfach deutlich worden. Aus diesem Grund hat die IT-Industrie inzwischen viele ergänzende bzw. alternative Vorgehensmodelle entwickelt (s. z. B. <http://de.wikipedia.org/wiki/Wasserfallmodell>).

Der Gedanke, ein Softwareprojekt als pädagogisches Projekt in der Schule durchzuführen, wurde von Wolfgang Ambros ganz grundsätzlich kritisiert:

Es ist geradezu absurd, unter Berufung auf das pädagogische Projekt der Arbeitsschule [...] einen Berufszweig zu nehmen, zu dessen Wesen knallharte Konkurrenz, Kalkulation und Leistungsausbeutung gehört. Soziale und selbstbestimmende [sic!] Lernkomponenten in der pädagogischen Schutzhütte und industrielle Effektivität in der Wettbewerbssituation sind unvereinbar wie Feuer und Wasser ([BZ92], S. 194).

Gleichwohl sieht auch Wolfgang Ambros eine Reihe guter Gründe, „softwaretechnologische Methoden“ in den Informatikunterricht einzubeziehen. Auch Projekte im Informatik-Unterricht erachtet er als sinnvoll, allerdings nicht als Simulation eines professionellen Software-Projekts. Es geht Ambros also eher um die Frage des *wie* als um die Frage des *ob* (ebd., S. 195f.).

In den nachfolgenden Berliner Informatik-Rahmenplänen, beginnend mit [S93], ist die Ausrichtung an dem Wasserfallmodell deutlich zurückgenommen, Software-Projekte und Software-Engineering sind aber weiterhin wichtige Methoden und Inhalte. Inzwischen wird auch diskutiert, Aspekte von eXtreme Programming (XP) – dem Gegenentwurf zum Wasserfallmodell (s. z. B. http://de.wikipedia.org/wiki/Extreme_Programming) – als Methode im Informatikunterricht einzusetzen [L08].

Der Berliner Rahmenplan von 1985 sah Projekte nur für das dritte Lernjahr vor:

Allerdings ergibt sich erst für die fortgeschrittenen Schüler die Möglichkeit, Probleme der Anwendung, Algorithmik, Rechnerorganisation und Rechnerbedienung bei der Bearbeitung realitätsnaher Projekte integriert und in ihrer tatsächlich auftretenden Relevanz zu erfahren und dabei in heute typische Methoden zur Herstellung komplexer Software eingeführt zu werden ([KS84], S. 135).

Für die Organisation der Projektarbeit wurden mehrere Alternativen vorgesehen:

- ein ganzjähriges Projekt (diese Variante wird in [JKS83] empfohlen),
- zwei halbjährige Projekte,
- ein halbjähriges Projekt und im anderen Halbjahr ein selbstgewähltes Thema.

In jedem Fall war also ein mindestens halbjähriges Projekt verpflichtend.

Im Plan aus dem Jahr 1993 [S93] wurde die Möglichkeit eines ganzjährigen Projektes zurückgenommen. Es war nur noch für das dritte Semester ein halbjähriges Projekt vorgesehen, für das vierte Semester wurden alternativ vier mögliche Vertiefungsgebiete vorgeschlagen:

- Theoretische Informatik: Automaten
- Nicht-prozedurale Sprachkonzepte (logisch oder funktional)
- Datenbanken
- Computergrafik

Diese Vertiefungsgebiete (insbesondere die beiden letzten) konnten durchaus auch projektorientiert bearbeitet werden. Die Behandlung eines selbstgewählten Themas (z. B. Technische Informatik) war auf Antrag weiterhin möglich.

1.5 Zwischenbilanz

Vor 25 Jahren war der Berliner Rahmenplan Informatik für die SekII gekennzeichnet durch die Betonung des anwendungsorientierten Ansatzes sowie durch ein verpflichtendes Softwareprojekt. Vom zeitlichen Umfang her lag der Schwerpunkt auf der Vermittlung von Algorithmen und Datenstrukturen. Demgegenüber war die Fachsystematik der Kerninformatik (technische, theoretische und praktische Informatik) für den Rahmenplan nur von untergeordneter Bedeutung.

Das ist aus Sicht der Autoren auch gut so, weil eine stark ausgeprägte Orientierung an der Fachsystematik immer die Gefahr impliziert, die Inhalte des Schulfaches als verkleinertes Abbild der Hochschuldisziplin zu verstehen.

Didaktik soll in der Regel das aus den jeweiligen Fachwissenschaften herausfiltern, das von allgemeiner, existentieller Bedeutung für das Leben (Überleben und menschenwürdiges Leben) ist und daher als lehrnotwendig legitimiert gelten kann. (Wolfgang Hilligen 1991). Ein Missverständnis von Didaktik ist die so genannte „Abbilddidaktik“, nach der es nur darum gehe, die Ergebnisse der Fachwissenschaft auf welche Weise auch immer in die Schülerköpfe zu transportieren. Die Fachwissenschaften sind nur ein Bezugspunkt der Didaktik, andere sind die Gesellschaft und die Bedürfnisse des Schülers selbst (<http://de.wikipedia.org/wiki/Didaktik>).

Beispiele für die Abbilddidaktik finden sich in vielen Fächern. Hier soll nur die „Wissenschaftsorientierung“ in der Mathematik“ aus den 60er und 70er Jahren des letzten Jahrhunderts als prominentes Beispiel genannt werden. Wenn man Lehrbücher aus dieser Zeit in die Hand nimmt, erscheinen sie häufiger abstrakter (und abschreckender) als einführende Universitätsliteratur.

Aber auch in der Informatik gibt es vereinzelt Verfechter der Abbilddidaktik. So heißt es bei Christian Maurer:

Unabdingbare Voraussetzungen für die Abbildung wissenschaftlicher Erkenntnisse auf Schulunterricht sind realistische Einschätzungen der Machbarkeit im Kontext des Bildungs- und Erziehungsauftrages der Schule – im Klartext: scharf durchdachte didaktische Reduktionen wesentlicher fachlicher Inhalte. [...] Für den Informatikunterricht in der gymnasialen Oberstufe sind in erster Linie die Kerngebiete eines Grundstudiums relevant [M08].

Hiermit ist offenbar nicht nur die (selbstverständliche) Forderung gemeint, dass schulisches Lernen nicht in Widerspruch zum jeweiligen wissenschaftlichen Erkenntnisstand geraten darf und Begriffsbildungen so angebahnt werden müssen, dass sie kein Lernhindernis für die weiterführende Beschäftigung werden. Insofern ist eine wissenschaftlich solide Lehrerausbildung unverzichtbar. Beim zitierten Ansatz stellt sich allerdings die Frage, ob die Gleichsetzung von schulischem und universitärem Lernen eine optimale Wissenschaftspropädeutik darstellt. Zumindest ist nicht erkennbar, wie die oben erwähnten anderen Bezugspunkte (die Gesellschaft und die Bedürfnisse des Schülers) angemessen berücksichtigt werden.

2 Was danach geschah

Im Jahr 2000 begannen Vorarbeiten für einen neuen Berliner Rahmenplan Informatik für die Sek II. Anlässe waren u. a. die geplante Einführung des Leistungskurses Informatik, neuere fachdidaktische Entwicklungen (insbes. die Bedeutung der Objektorientierung im Informatik-Unterricht) sowie die Auswertung der Erfahrungen mit dem bis dahin gültigen Rahmenplan [S93].

Zur Auswertung der Erfahrungen wurden in den Jahren 2001 und 2002 auf den Landestagungen der GI-Fachgruppe IBBB (Informatik-Bildung Berlin Brandenburg) zwei Workshops zu dem geplanten neuen Rahmenplan durchgeführt.

Im Schuljahr 2003/04 wurde von der Senatsschulverwaltung die Rahmenplankommission Informatik SekII eingesetzt, die bereits am 23. 5. 2004 einen Entwurf des neuen Rahmenplans veröffentlichen und zur Diskussion stellen konnte. Dieser Rahmenplan wurde nach Überarbeitung zum Schuljahr 2005/06 in Kraft gesetzt und erhielt dabei aus Gründen, die hier nicht weiter interessieren, den etwas ungewöhnlichen Namen „Curriculare Vorgaben“ [S05].

2.1 Curriculare Vorgaben Informatik

Was war neu an den curricularen Vorgaben?

Eine wichtige Änderung, die allerdings nicht nur die Informatik betraf, war die Umstellung von den durch den Behaviorismus geprägten Lernzielen auf Kompetenzen, die dem konstruktivistischen Ansatz in der Lerntheorie verpflichtet sind. Kompetenzen werden verstanden als individuelle Disposition zur Lösung von Problemen in variablen Situationen (vgl. [WE01]), während Lernziele punktuelle Verhaltensänderungen in einzelnen Lehr- und Lernprozessen beschreiben. Wenn mitunter operationalisierte Lernziele mit

„Kompetenzprofilen“ gleichgesetzt werden (s. [M08]), zeigt dies, dass dieser grundlegende Unterschied offenbar noch nicht überall verstanden wurde.

In den Curricularen Vorgaben wurden erstmals Eingangsvoraussetzungen für die Qualifikationsphase und Abschlusstandards für das Abitur in Informatik formuliert. Standards sind dabei für eine bestimmte Jahrgangsstufe konkretisierte Kompetenzen. Mit der Beschränkung auf Kerninhalte wurden den schulischen Fachkonferenzen vielfältige Möglichkeiten zur selbstgestalteten Schwerpunktbildung eingeräumt.

Veränderungen ergaben sich

- bei der Gestaltung des Einstiegsunterrichts; hier wurden neben dem bisher einzig möglichen Einstieg über die Dekonstruktion eines mäßig komplexen Softwaresystems alternative Zugänge ermöglicht (internetorientierter Zugang, Einstieg über eine Minisprache, [S05], S. 11) sowie
- durch die freie Wahl bei der Anordnung von Projektsemester und Vertiefungsgebiet im dritten Lernjahr. Hierdurch wurden z. B. Projekte mit Datenbanksystemen ermöglicht.

Auf diese Weise wurde den Schulen weitgehende Freiheit bei der Erarbeitung schul-eigener Curricula ermöglicht (Einstiegswege, Vertiefungsgebiete, Projektthemen).

Leider galten die Curricularen Vorgaben nur für die zwei Schülerjahrgänge, die 2008 bzw. 2009 das Abitur abgelegt haben. In im Frühjahr 2004 beschlossen die Bundesländer Berlin, Brandenburg und Mecklenburg-Vorpommern, für alle Fächer der Sek II Kerncurricula für die Qualifikationsphase zu entwickeln. Damit waren die Curricularen Vorgaben der letzte eigenständige Berliner Rahmenplan Informatik SekII.

2.1 Länderübergreifendes Kerncurriculum Informatik

Im September 2004 fiel der Startschuss für das Projekt eines 3-Länder-Kerncurriculums. Die Federführung für die einzelnen Curricula wurde auf die drei Bundesländer verteilt, für Informatik fiel sie auf Brandenburg. Aus diesem Grunde konnte Berlin nur einen Vertreter für die länderübergreifende Kommission entsenden.

In Log in berichtete Manfred Vollmost ausführlich über das Kerncurriculum [V05]. In Berlin gilt es als Teil des Rahmenlehrplans für die gymnasiale Oberstufe für alle Schülerinnen und Schüler, die seit dem Schuljahr 2006/07 den Informatikunterricht aufgenommen haben [S06].

Ein länderübergreifendes Kerncurriculum zu schaffen ist ein ambitioniertes Vorhaben, das grundsätzlich zu begrüßen ist, weil es der Zersplitterung des Bildungswesens in der Bundesrepublik Deutschland entgegenwirkt. Aber selbst wenn nur drei Bundesländer beteiligt sind, unterscheiden sie sich doch zum Teil gravierend in den organisatorischen Vorgaben für die gymnasiale Oberstufe, von unterschiedlichen (Unterrichts-)Traditionen einmal ganz abgesehen.

Für die Qualifikationsphase werden im Kerncurriculum sechs fachbezogene Kompetenzbereiche definiert:

Neben den Kompetenzen, die sich eher auf den Erwerb und die Anwendung von Inhalten beziehen (*Informatiksysteme verstehen – mit Informationen umgehen – Wechselwirkung zwischen Informatiksystemen, Menschen und Gesellschaft beurteilen*), gibt es Kompetenzen, die verstärkt prozessorientiert ausgerichtet sind (*Problemlösen – Kommunizieren und Kooperieren*) und die im Informatikunterricht eine besondere Ausprägung erfahren. Die Kompetenz des *informatischen Modellierens* umfasst sowohl inhalts- als auch prozessbezogene Aspekte ([S06], S. 10, Hervorhebungen durch die Autoren).

Für den Unterricht der Qualifikationsphase werden für die genannten Kompetenzbereiche sowohl Eingangsvoraussetzungen als auch die angestrebten abschlussorientierten Standards für Grund- und Leistungskursniveaus benannt. Danach werden fünf inhaltlich orientierte Themenfelder für den Unterricht beschrieben: Datenbanken – Rechner und Netze – Softwareentwicklung – Sprachen und Automaten – Informatik, Mensch und Gesellschaft.

Das Kerncurriculum lässt durch eine gewisse Offenheit alternative Realisierungen durch schulinterne Curricula zu, droht aber durch die separate Auflistung der Inhalte das eigentlich neue, nämlich den Kompetenzansatz aus den Augen zu verlieren. Ein weiteres Problem ist die Stofffülle. Der Anspruch, ein **Kerncurriculum** zu sein, das nur 60% der zur Verfügung stehenden Unterrichtszeit abdeckt, kann damit sicherlich nicht erfüllt werden.

Aus Berliner Sicht kamen mit dem Kerncurriculum folgende verpflichtende Inhalte neu hinzu: Datenbanken (bisher nur propädeutisch in der Einführungsphase bzw. als Vertiefungsgebiet), Sprache und Automaten für den Grundkurs sowie für den Leistungskurs zusätzlich Grammatiken, formale Sprachen und Turingmaschine oder Registermaschine.

Im Kerncurriculum werden keinerlei Kurssequenzen angegeben. Dies ist konsequent im Sinne eines kompetenzorientierten Ansatzes, der nur die zu erreichenden Abschlussstandards beschreibt, den Weg dahin aber offen lässt. Leider wurde dies von den Bildungsverwaltungen als mögliche Überforderung der Lehrenden angesehen, so dass in Berlin und Brandenburg im länderspezifischen Teil noch jeweils unterschiedliche Kursfolgen formuliert wurden. In Mecklenburg-Vorpommern gibt es auch für Informatik ein Zentralabitur, dort wird eine detailliertere Steuerung durch abiturbezogene Vorgaben und Beispielaufgaben realisiert.

Für Berlin setzen die je vier Grund- und Leistungskurse gleiche Schwerpunkte und haben gleiche Titel: Datenbanken und Softwareentwicklung I – Datenbanken und Softwareentwicklung II – Grundlagen der Informatik und Vertiefungsgebiet – Softwareprojekt. Die Leistungskurse sind eine inhaltliche Obermenge der Grundkurse. Diese Anordnung würde grundsätzlich die für viele Schulen attraktive Lösung bieten, den Leistungskurs durch Grundkurs und einen Ergänzungskurs zu realisieren. Hiervon kann jedoch aufgrund der Vorgaben durch den Schulsenat z. Zt. (leider) noch kein Gebrauch gemacht werden.

Die Beschreibung der Kurse der Qualifikationsphase beschränkt sich auf die jeweils zu behandelnden Inhalte. Durch die ausschließliche Betonung der Inhaltsbeschreibungen, die am ehesten bei der konkreten Unterrichtsplanung als Referenz zu Grunde gelegt werden, wird abermals die Kompetenzorientierung konterkariert, stattdessen werden eher die schon totgeglaubten Stoffverteilungspläne wiederbelebt. Offenbar wird dem kompetenzbasierten Ansatz nicht so recht getraut.

Die aufgegriffenen Inhalte stellen eine der Entwicklung der Informatik und ihrer Anwendungen entsprechende Aktualisierung dar. Allerdings müsste aus Sicht der Autoren der Umfang der auch im Grundkurs verpflichtenden Themen der theoretischen Informatik hinterfragt werden.

Positiv im Berliner Rahmenlehrplan ist aus Sicht der Autoren das Weiterbestehen des Projektsemesters zu vermerken. Das Softwareprojekt ist aber nicht Bestandteil des Kerncurriculums und in Brandenburg und Mecklenburg-Vorpommern leider auch kein verpflichtender Teil des Informatikunterrichts der Sek II.

Für den der Qualifikationsphase vorausgehenden Unterricht (also Basis- und Profilkurs), der im Kerncurriculums nicht beschrieben wird, werden für Berlin im wesentlichen die curricularen Vorgaben übernommen.

3 Fazit

Was hat sich also im Kern in den letzten 25 Jahren geändert? Die schon positiv vermerkte inhaltliche Modernisierung im Rahmenlehrplan (Datenbanken, Objektorientierung) sollte eigentlich selbstverständlich sein. Die Änderung von Lernzielen zu Kompetenzen betrifft alle Fächer, nicht nur die Informatik. Allerdings erfolgte diese Umstellung, wie bereits erwähnt, nur halbherzig.

Die Idee der leistungs- und wissenshomogenen Lerngruppe scheint immer noch Grundlage bzw. Vorgabe der Lehrplankonstruktion zu sein:

Im 1. Halbjahr der Einführungsphase sind ggf. unterschiedliche Vorerfahrungen bei den Schülerinnen und Schülern durch geeignete Maßnahmen der Binnendifferenzierung zu berücksichtigen. [...] Damit soll sichergestellt werden, dass sie am Unterricht in der Qualifikationsphase erfolgreich teilnehmen können, ([S06], S.V)

Das Thema „Anwendungen und Auswirkungen“ findet sich – zumindest teilweise – im Kompetenzbereich „Informatik, Mensch und Gesellschaft“ wieder. Eine zeitgemäße Weiterentwicklung der Anwendungsorientierung stellt das Projekt „Informatik im Kontext (IniK)“ dar (vgl. <http://www.informatik-im-kontext.de/>).

Der Themenbereich Algorithmik wird inzwischen eingebettet in die objektorientierte Modellierung behandelt. Die schwierige fachdidaktische Frage, ob zuerst in die „kleine“ Algorithmik oder in die objektorientierte Modellierung eingeführt werden soll oder ob beides integriert behandelt werden sollte, wird durch den Rahmenlehrplan nicht vorentschieden.

An der Wertigkeit von Rechnerorganisation und Rechnerbedienung hat sich in den letzten 25 Jahren nicht allzuviel geändert, auch im Kerncurriculum hat die technische Informatik keinen herausgehobenen Stellenwert.

Das verpflichtende Softwareprojekt hat sich als Konstante der Berliner Rahmen(lehr-)pläne erhalten, im zeitlichen Umfang hat es allerdings deutlich abgenommen. Durch die Stofffülle, die auch in der Informatik inzwischen erheblich zugenommen hat, ist das oben zitierte „Korsett“ des Rahmenlehrplans spürbar enger geworden.

Leider ist die Informatik auch in einem weiteren Punkt den traditionellen Fächern gefolgt: Die Schülerin, der Schüler wird immer noch als Objekt der Unterrichtsbemühungen des Lehrers gesehen, nicht als Subjekt ihres bzw. seines Lernprozesses. Die Frage „Wirst du noch unterrichtet oder lernst du schon?“ muss daher aus Schülersicht weitestgehend zu Ungunsten des selbstgesteuerten Lernens beantwortet werden.

Literaturverzeichnis

- [A78] Arlt, Wolfgang (Hrsg.): EDV-Einsatz in Schule und Ausbildung. München: Oldenbourg, 1978.
- [AH84] Arlt, Wolfgang; Haefner, Klaus (Hrsg.): Informatik als Herausforderung an Schule und Ausbildung. GI-Fachtagung Berlin, Oktober 1984. Berlin, Heidelberg (Springer), 1984.
- [B96] Baumann, Rüdiger: Didaktik der Informatik. Stuttgart (Klett), 1996.
- [BZ92] Bosler, Ulrich; Ziebarth, Wolfgang (Hrsg.): Schul/Computer/Jahrbuch '93/94. Hannover (Metzler) und Stuttgart (Teubner), 1992.
- [GR76] Gruppe Informatik in Berliner Schulen (Beer/Koerber/Reker/Sack/Schulz): Stand der Curriculum-Entwicklung für den Informatik-Unterricht in Berlin. Berlin, 1976.
- [HMS99] Hampel, Thorsten; Magenheim, Johannes; Schulte, Carsten: Dekonstruktion von Informatiksystemen als Unterrichtsmethode – Zugang zu objektorientierten Sichtweisen im Informatikunterricht. In: [S99], S. 149-164.
- [JKS83] Jänichen, Stefan; Koch, Wilfried; Schürmann, Gerd: Software Engineering und Lehrerbildung im Fach Informatik. In: LOG IN 3 (1983), Heft 2, S. 25-29.
- [KS84] Koch, Wilfried; Sack, Lothar: Konzepte und Inhalte für einen Lehrplan „Informatik“ in der gymnasialen Oberstufe. In: [AH84], S. 133-138.
- [L08] Leiser, Shenja: Adaption von Aspekten des Extreme Programming zur Steigerung der Schüleraktivität bei der Entwicklung eines Spiels mithilfe der Entwicklungsumgebung *Greenfoot*. Berlin (Staatsexamensarbeit), 2008.
[http://bscw.schule.de/pub/bscw.cgi/d567998/Examensarbeit%20Hr.%20Leiser%20\(2008\)](http://bscw.schule.de/pub/bscw.cgi/d567998/Examensarbeit%20Hr.%20Leiser%20(2008)).
- [M08] Maurer, Christian: Lerninhalte und Kompetenzprofile im Informatikunterricht der gymnasialen Oberstufe. <http://lwb.mi.fu-berlin.de/inf/mix/kompetenzen/index.html>
- [S72] Senator für Schulwesen: Vorläufiges Grundprogramm für das Fach Informatik. Berlin 1972.
- [S77] Senator für Schulwesen: Vorläufiger Rahmenplan für Unterricht und Erziehung in der Berliner Schule. Gymnasiale Oberstufe. Informatik. Berlin 1977.
- [S78] Sack, Lothar: Ein Modellversuch zum Informatikunterricht in der Sekundarstufe II. In: [A78], S. 56-64.

- [S85] Der Senator für Schulwesen, Berufsausbildung und Sport: Vorläufiger Rahmenplan für Unterricht und Erziehung in der Berliner Schule. Gymnasiale Oberstufe. Informatik. Berlin 1985.
- [S93] Senatsverwaltung für Schule, Berufsausbildung und Sport: Vorläufiger Rahmenplan für Unterricht und Erziehung in der Berliner Schule. Gymnasiale Oberstufe. Informatik. Berlin 1993.
- [S99] Schwill, Andreas (Hrsg.): Informatik und Schule – Fachspezifische und fachübergreifende didaktische Konzepte. Tagungsband der Infos'99, Berlin, Heidelberg (Springer), 1999.
- [S05] Senatsverwaltung für Bildung, Jugend und Sport Berlin: Curriculare Vorgaben für die gymnasiale Oberstufe – Informatik. Berlin 2005.
http://bscw.schule.de/bscw/bscw.cgi/d610953/Curriculare_Vorgaben_Informatik_SekII
- [S06] Senatsverwaltung für Bildung, Jugend und Sport Berlin: Rahmenlehrplan für die gymnasiale Oberstufe – Informatik. Berlin 2006. http://www.berlin.de/imperia/md/content/sen-bildung/schulorganisation/lehrplaene/sek2_informatik.pdf
- [SZ78] Schulz-Zander, Renate: Analyse curricularer Ansätze für das Schulfach Informatik (in [A78], S. 40-49).
- [V05] Vollmost, Manfred: Ein Kerncurriculum Informatik – Zur Diskussion gestellt. In: LOG IN 135 (2005), S. 54-~~22~~.
- [WE01] Weinert, F. E. (2001). Vergleichende Leistungsmessung in Schulen – eine umstrittene Selbstverständlichkeit. In: F. E. Weinert (Hg.): Leistungsmessungen in Schulen. Weinheim/ Basel, S. 17-31.

Unterricht im Wandel? – Das Potenzial der Informatik

Maria Knobelsdorf

Institut für Informatik
Freie Universität Berlin
Königin-Luise-Str. 24-26
14195 Berlin
maria.knobelsdorf@fu-berlin.de

Abstract: Dieser Artikel stellt Ergebnisse aus dem Forschungsprojekt *Computernutzungsbiographien* an der Freien Universität Berlin vor und analysiert, wie sich Schülerinnen und Schüler außerhalb des Informatik-Unterrichts mit Informatik und Computern beschäftigen, wie sie dabei selbstständig und kreativ agieren und sie ein Informatik-Unterricht anspricht, der dieses einbezieht und konkretisiert. Die vorgestellten Ergebnisse gewinnen durch die Debatte um die Zukunft der deutschen Schullandschaft im Zuge der PISA-Studie und Lernen in der Wissensgesellschaft maßgeblich an Bedeutung.

1 Einleitung

Seit der ersten PISA-Studie ist bekanntermaßen wieder einmal eine Diskussion über die Zukunft der deutschen Schullandschaft entbrannt. Im Zuge dieser Debatte und der sich daraus ableitenden Suche nach Veränderungen, ist ein vermehrtes Interesse über Schulen zu beobachten, die sich in den letzten zehn, zwanzig Jahren erfolgreich reformiert haben. Diese neu geschaffenen Ganztagschulen haben den 45-Minuten Rhythmus ebenso abgeschafft wie curricularen Unterrichtsablauf im Klassenverband. Die SuS (Schülerinnen und Schüler) lernen stattdessen jahrgangsübergreifend nach individualisierten Lernplänen und im Rahmen von Kompetenzmodellen. Offener Unterricht, Lernen durch Lehren, Projektarbeit und Präsentationen des Gelernten sind die dabei hauptsächlich anzutreffenden Lernformen. Ist das neuer Wein in alten Schläuchen? Solche Konzepte scheinen nicht wirklich neu zu sein, denn neben der traditionellen Schulpädagogik gab es immer die sog. Reformpädagogik, die unter anderem auf das lernende Subjekt fokussierte. Doch weder eine Montessori-Didaktik noch die kritische Auseinandersetzung über fremdbestimmtes Lernen und kooperative Lernformen in der didaktischen Literatur seit den siebziger Jahren, haben der traditionellen Unterrichtsform und dem von der Lehrkraft geführten Ganztagsunterricht etwas anhaben können.

An dieser Debatte ist neu, dass der PISA- und TIMMS-Vergleich diesen Schulen exzellente Ergebnisse bescheinigt, während die „traditionellen“ Schulen durch mittelmäßige Ergebnisse landesweit erschüttert wurden. Oft genug haben sich die heutigen „Vorzeigeschulen“ aus reiner Not zu reformieren begonnen, weil sie den „normalen“ Betrieb nicht mehr bewältigen konnten. Leistung, Umgang mit Vielfalt und Schule als lernende Institution sind in diesen Schulen die zentralen Säulen einer SuS-orientierten Unterrichtsgestaltung. Sie gehen einher mit moderner Schulpädagogik und pädagogischer Psychologie, die jedem Schüler und jeder Schülerin zugestehen anders und im unterschiedlichen Tempo zu lernen, mit unterschiedlichen Interessen und Voraussetzungen ihrem Lernprozess zu begegnen. Der Deutsche Schulpreis trägt dieser Entwicklung und Reformfreude deutlich Rechnung [RBS].

Das Bundesministerium für Bildung und Forschung begrüßt diese Entwicklung und fördert diese Pädagogik der Vielfalt z.B. durch Ausbau von Ganztagschulen [BMPF]. Eine so breit angelegte Transformation der pädagogischen und didaktischen Konzepte, verlangt den Beteiligten jedoch viel ab: LuL (Lehrerinnen und Lehrer), die maßgeblich ihre Lehrressourcen investiert haben, um im bestehenden System Unterrichtsreihen zu entwickeln, sollen nun anscheinend diese verwerfen und zu „Moderatoren“ des Unterrichts werden, die SuS nicht mehr unterrichten, sondern in ihrem individuellen Lernprozess „begleiten und coachen“. SuS, die sich nach Jahren an bestimmte Unterrichtsformen gewöhnt und daran angelehnte Aneignungsmethoden gelernt haben, sollen sich nun anscheinend selbst Lernziele setzen und, sich selbst oder gegenseitig unterrichten und aktiv ihren eigenen Lernprozess gestalten.

Wo steht in dieser ganzen Debatte und Entwicklung das Unterrichtsfach Informatik? Die Bildungsstandards für die Informatik sind veröffentlicht [GSIS], die Notwendigkeit einer informatischen Bildung wird immer mehr erkannt und dennoch genießt das Schulfach Informatik ein gewisses „Schattendasein“ an der Schule. Die Etablierung der Informatik in der Schule und der Informatikdidaktik in der Hochschule, sowie die weitere Entwicklung von Bildungsstandards und Kompetenzmodellen für die Informatik stellt genug Herausforderungen an die Fachcommunity und generelle Debatten über didaktische Ansätze und sog. Paradigmenwechsel entziehen sich dem Fokus. Doch gerade die Informatik hat als nicht alt eingesessenes Unterrichtsfach, deren Fachwissenschaft selbst sich extrem schnell weiterentwickelt, ein Potenzial zu bieten, das LuL und SuS nutzen können, um diesen Paradigmenwechsel wirkungsvoll gemeinsam umzusetzen.

Der hier vorgelegte Artikel versucht im nächsten Abschnitt dieses Potenzial aus Lerner-Perspektive zu rekonstruieren. In Abschnitt 3 wird der angesprochene Paradigmenwechsel kurz skizziert, um im Abschnitt 4 einen Ausblick auf den Informatik-Unterricht (IU) zu geben.

2 Lernprozesse in Computernutzungsbiographien

Im Forschungsprojekt *Computernutzungsbiographien* an der Freien Universität Berlin wird biographisches Lernen im Kontext der Informatik untersucht [KS07], [Kn09]. Hierbei geht es nicht generell um Lernen, sondern Lernen von und mit informatischen Artefakten im biographischen Kontext. Dazu wurde ein Datenerhebungsinstrument entwickelt, das autobiographische Texte über Computernutzungserfahrungen von Studienanfängern liefert. Personen werden gebeten ihre Erfahrungen und Erlebnisse mit dem Computer in Form einer Autobiographie niederzuschreiben und ermutigt mit dem ersten Computerkontakt den sie erinnern zu beginnen. Der Schreibprozess wird mit sogenannten Locktexten (Ausschnitten aus anderen Biographien) angeregt. Die Schreibaufforderung ist offen gestaltet, um den Personen die Möglichkeit zu lassen, zu entscheiden über welche Erlebnisse sie wie berichten. So sind Computernutzungsbiographien von Informatikstudienanfängern gleichzeitig Geschichten, die vermitteln wie und warum die Studierenden zur Informatik gekommen sind.

Die schriftlich erhobenen Daten sind Teil eines qualitativen empirischen Forschungsansatzes und wurden zu Beginn des Projekts mit der Analysemethode des Theoretical Samplings aus der Grounded Theory untersucht [Bö03]. Im weiteren Verlauf wurde Qualitative Inhaltsanalyse nach Mayring [Ma07] angewendet unterstützt durch die Kodierungs- und Analysesoftware MaxQDA [Ku07].

Den Hauptuntersuchungsrahmen bilden vier Dimensionen: Weltbild, Selbstbild, Handlungsweisen und Prozess. Diese vier Dimensionen beschreiben verschiedene Aspekte, die im biographischen Lernprozess entstehen. Das Selbstbild entsteht durch emotionale und intentionale Lebensprozesse, die bezogen auf physische, psychische und soziale Momente die subjektiven Persönlichkeitsbildung und Selbstreflektion prägen. Im Forschungsprojekt liegt der Fokus auf Computernutzungserfahrungen und der Selbstorientierung in der informatischen Welt. Im Weltbild des Subjekts konstruiert sich ein ihm sinngebendes Verhältnis zur sachlich-sozialen Welt gesellschaftlicher Bedeutungszusammenhänge (informatische Artefakte und ihre gesellschaftliche Bedeutungsstruktur). Die Handlungsweisen schließen Formen und Strategien von Verhalten, Reaktion und Lernen ein, während die zeitliche Dimension und der Wandel der anderen Dimensionen als Prozess abgebildet wird ([KS07], 5 ff.), [Kn09].

Dieser Artikel befasst sich mit der Dimension Handlungsweisen und Lernformen. In mehreren Studien mit Biographien von Informatik und Bioinformatik Studienanfängerinnen und -anfängern konnten immer wieder selbst angeleitete Arbeits- und Lernprozesse abgeleitet werden, die Erforschen und Experimentieren am Computer ebenso beinhalteten wie zahlreiche Aspekte der Kreativität [KR08]. So stellt jede Computernutzungsbiographie gleichzeitig einen kurzen Abriss der eigenen Lerngeschichte in Bezug auf Computer und Informatik dar, wie der folgende Textausschnitt illustriert:

„Eines Tages las ich ein Buch in dem es um Computer und Programmierung ging. Obwohl dieses Buch kein Sachbuch, sondern mehr ein Kinderroman war, enthielt es einige Zeilen der Computersprache BASIC. Ich fand heraus, dass auf unserem Computer ein Basic-Interpreter installiert war und fand in unserem Bücherregal ein Buch über Basic-Programmierung. Ich spielte ein bisschen mit dieser Programmiersprache herum und obwohl ich von der Idee eigenen Programme zu schreiben, über alle Maßen fasziniert war, hatte ich nicht den Elan mehr als etwas triviales wie z.B. einen simplen Taschenrechner zu schreiben.“

2.1 Außerschulische Lernaktivitäten und –formen

In der vorliegenden Untersuchung mit 189 Studienanfängerinnen und -anfängern der Informatik und Bioinformatik wurden Lernaktivitäten untersucht. Die geschilderten Lernaktivitäten können vereinfacht in drei Gruppenaufgeteilt werden: Selbstlerner, Fremdlerner und Beobachterlerner. Die Beobachterlerner lernen, indem sie jemand Dritten (z.B. ein Familienmitglied) bei einer Tätigkeit beobachten und das Beobachtete anschließend imitieren. Diese Form von Lernen findet hauptsächlich am Anfang der Biographie statt und meistens wenn die Personen jüngeren Alters sind:

„Schon im Alter von 2-3 Jahren interessierte mich der Computer meines Vaters, der bei uns im Keller stand sehr. So saß ich oft neben ihm und schaute ihm dabei zu, wie er den Computer nutzte z.B. für Textverarbeitung aber auch zum Spielen. Mit 6 Jahren bediente ich den Computer bereits selbst.“

Die Gruppe der Selbstlerner leiten sich beim Lernen selbst an und bestimmten wann und was sie lernen. Das geht vom einfachen ausprobieren des vom Computer Dargebotenen, über learning-by-doing, bis hin zu Lernprozessen, die bewusst geplant, organisiert und durchgeführt werden. Manche kämpfen sich eher planlos durch ohne ein tiefgehendes Verständnis zu entwickeln, andere planen gezielt, holen sich bei Bedarf fachliche Hilfe oder Materialien. Sehr oft schließen sich manche mit Gleichgesinnten zu „Lerngruppen“ zusammen. Diese Form des Lernens ist neben den Schilderungen zum IU die am häufigsten und am ausführlichsten geschilderte Phase. Die folgenden Textbeispiele dienen als Beispiel:

„Durch Probieren, lernte ich Funktionsweisen, und vor allem den Umgang mit dem PC.“

„Hier entdeckte ich einerseits die 'Killerspiele' von damals und andererseits die DOS Shell, die zu meinem liebsten Spielzeug wurde. Ich lernte alle Befehle in und auswendig und erforschte das System, spielte in allen Konfigurationsdateien herum und schrieb Batchdateien um so viel wie möglich zu automatisieren.“

„In der Schule tat ich mich mit Freunden zusammen, und wir setzten uns mit PHP, JavaScript, HTML und C++ auseinander.“

Doch nicht jede Person beherrscht das Selbstlernen. Manche schreiben sogar direkt, dass sie gerne selbst gelernt hätten, aber dies nicht umsetzen konnten, wie das folgende Beispiel illustriert:

„Vor dem Informatik-Unterricht an meiner Schule hatte ich noch keine Programmiererfahrungen und hätte auch keine Ahnung gehabt, wie ich es mir selbst hätte beibringen können.“

Die Fremdlerner lernen durch jemand Dritten, der sie anleitet und unterrichtet. Das kann z.B. der IU oder ein Programmierkurs an der Volkshochschule sein. Entscheidend daran ist, dass auch die Fremdlerner sich bewusst dafür entscheiden, wann sie was lernen möchten.

Selbstverständlich gibt es aus diesen drei Gruppen, Mischformen und diese sind in den Biographien am häufigsten zu beobachten. Viele berichten von Anfängen als Beobachterlerner, einem Wechsel zum Selbstlernen und wie sie später zum Ende der Sek. I im IU auch durch Fremdlernern ihr Wissen erweitern konnten. Das Interessante hierbei ist, dass die Lernphase vor dem ersten IU meistens eine Selbstlernphase ist, die sich über einen langen Zeitraum gestaltet und so prägend bezüglich selbstbestimmten Entscheidungen über Lerninhalte und -ziele (wenn auch im nicht institutionellen Kontext) ist.

2.2 Erwartungen an den Informatikunterricht

Da Informatik kein Pflichtfach ist, haben alle SuS, die IU in der Schule wählen konkrete Erwartungen was und auch wie sie im Unterricht lernen wollen. Ob nun eine grundlegende Einführung in die PC-Bedienung und Wartung oder eine Vorbereitung auf das Informatikstudium erwartet wird, SuS wenden sich enttäuscht ab, wenn sie im IU nicht genau an ihrem Vorwissen anknüpfen können:

„Mein Interesse begann an der Schule. Mit einem freien Wahlkurs. Ich war sofort fasziniert. Ich konnte mir den Grund nicht erklären doch es war sehr unterhaltsam etwas eigenes auf dem PC zu erstellen auch wenn es nur in BASIC war. In höheren Klassen an festen Kursen Informatik nahm das Interesse ab. Das lag daran das ich in der Schule erklärt bekommen hab wie Word und Excel funktioniert. Dabei war ich selbst bereits viel weiter.“

Wenn beispielsweise ein Schüler den PC zu bedienen gelernt hat, will er vielleicht als nächstes lernen, wie der Computer aufgebaut ist, während er mit *objektorientiertem Modellieren* nichts anfangen kann. Eine andere Schülerin hat bereits mit Basic und Pascal zu Hause experimentiert und kleine Programme verfasst und interessiert sich nun brennend für *OOP*, während sie das *Von-Neumann-Rechnermodell* wenig reizt:

„Langsam erkannten wir, dass man Spiele ja auch selbst schreiben konnte und belegten deswegen in der 11. Klasse den Informatikunterricht. Leider war unser Lehrer nicht der Meinung, dass wir im gesamten ersten Halbjahr einem etwas am Computer lernen sollten und so quälte er uns mit Binär- und Hexadezimal Zahlen, erklärte uns ausführlichst den „Von-Neumann-Rechner“ und lies uns schmoren. Deshalb suchten wir uns irgendwann Tutorials zu Pascal aus dem Netz und lernten das Programmieren so. Dadurch konnte der Lehrer uns allerdings auch nichts mehr beibringen, als wir endlich Turbo Pascal im Unterricht behandelten.“

Andererseits werden offene Lernformen als besonders positiv dargestellt:

„Am meisten hatte mir in der 13ten Klasse 4. Semester das freischaffende Projekt gefallen, bei dem wir selbst ein Programm in Teamarbeit programmierten.“

„Das Highlight meines bisherigen Informatikstudiums war das sog. Software-Praktikum in der 12. Klasse, wo wir 3 Programme zu programmieren hatten und 4 Monate auf uns allein gestellt waren und neue Inhalte selbst erlernen mussten.“

Diese und andere Biographien berichten vom gelungen IU mit engagierten LuL, die sich vom klassischen Unterrichtsablauf lösen und ihren SuS durch Projektarbeit über einen längeren Zeitraum die Möglichkeit zu selbstorganisiertem Lernen geben. Doch in welchem Kontext ist selbstorganisiertes Lernen eigentlich zu verstehen? Geht es einfach nur darum mehr Gruppen- und Projektarbeit zuzulassen und muss man nicht dann befürchten, aus dem IU wird eine Informatik-AG? Darum soll es im nächsten Kapitel gehen.

3 Selbstorganisiertes Lernen in der Forschung

Die in den Biographien geschilderten Lernformen des selbstorganisierten oder selbstbestimmten Lernens sind Ausprägungen einer Lernform, die seit langem in der Reformpädagogik aber seit kurzem auch in der pädagogischen Psychologie erforscht wird [He85], [WWE03], [Wo00], [Wi04]. Dabei werden verschiedene Begriffe synonym verwendet oder im jeweiligen Kontext unterschiedlich akzentuiert. Im Vordergrund steht die Selbstbestimmtheit des Individuums, was sich in den unterschiedlichen Begriffen widerspiegelt: selbstbestimmtes, selbstorganisiertes, selbsttätiges, selbstgesteuertes, selbstreguliertes, selbstkontrolliertes Lernen. Individuelle Interessen, Eigenmotivation und Entscheidungsmöglichkeiten des Subjekts sollen stärker berücksichtigt werden.

In ihrer Studie zum Lernen in der Freizeit von Schulkindern konstatieren Furtner-Kallmünzer et. al., dass Lernen in der Wissensgesellschaft bedeutet „dass jeder in Zukunft in der Lage sein muss, sich selbst in jeder Lebensphase das theoretische und praktische Wissen zu besorgen, das er benötigt, um in seiner Lebenswelt handlungsfähig zu bleiben. Dabei wird es dem Einzelnen überlassen, auf welche Weise er sich das jeweils notwendige Wissen beschafft, ob er dazu formale Schulungen in Anspruch nimmt oder informell im privaten oder beruflichen Alltag lernt“ ([FHJ03], S. 114).

Bannach legt dar, dass „Schule und Unterricht in der Gefahr stehen, sich zu einseitig auf die Seite der Erfüllung gesellschaftlicher Zwecke und Funktionen zu stellen. Des Weiteren räumen sie den Regeln und Verkehrsformen der Institution einen zu großen Stellenwert gegenüber den Handlungs- und Lernbedürfnissen der Schüler ein [...]“ ([Ba02], S. 349). Das Konzept des gegenwärtigen Schulsystems sieht die Selbstbestimmung der SuS so nicht vor. Durch Schulpflicht und vorgegebene Lerninhalte hat Schule leider einen Zwangscharakter. Doch auch in diesen eng vorgegebenen Strukturen können selbstorganisierte Lernprozesse ermöglicht werden. Bannach schlägt dazu die Themenarbeit als Umsetzungsmöglichkeit vor: „Die Ermöglichung von selbstbestimmten und selbstorganisiertem Lernen der Schüler an Interessenthemen kann als Gegengewicht zum traditionellen Unterricht betrachtet werden. Im Gegensatz zum rezeptiven, weitgehend passiven Lernen des herkömmlichen Unterrichts werden die Schüler in der Themenarbeit aufgefordert, Inhalte selbst zu bestimmen, sich selbst Ziele zu setzen und eigene Vorhaben selbstorganisiert und selbstverantwortlich voranzubringen“ ([Ba02], S. 350).

In dieser Diskussion um selbstorganisiertes Lernen schwingt auch ein Paradigmenwechsel mit: „ ‚Out‘ sind Institutionen, Lernziele und Lehre; ‚in‘ sind informelle Lernformen, Selbsttätigkeit und Medienarrangements – ganz zu schweigen von den Versprechen der Informationstechniken, der Multimedia und des Internet“ ([Fa99], 24 ff). Wenn hier also auch von Paradigmenwechsel die Rede ist, wogegen grenzen sich diese Begriffe dann genau ab? Selbstorganisiertes, selbstbestimmtes Lernen wird vereinfacht als Gegensatz zu fremdorganisierten, fremdbestimmten Lernen konzipiert. Aber Lernen ist doch auch eine kognitive Tätigkeit und somit an sich eine Form der *Selbsttätigkeit* an. Was genau ist also gemeint? Die hier stattfindende Abgrenzung lässt sich leichter nachvollziehen, wenn man sich kurz bewusst macht welchen wissenschaftlichen Standpunkt die akademische Psychologie vertritt und auf welcher Grundlage Lerntheorien entstanden sind, von denen sich letztendlich das Selbstlernkonzept abgrenzt.

3.1 Der Mensch als Bohnenpopulation

Die entstehende Psychologie im 19. Jahrhundert hatte eine noch gänzlich andere Herangehensweise an das Subjekt seiner Forschung (den Menschen also), als man das von der Psychologie des 20. Jahrhunderts gewohnt ist ([Ho84], S. 21). In seinem Grundriss der Psychologie legt Wundt (1897) dar, dass zwar sowohl die Naturwissenschaften als auch die Psychologie untersuchen, was wir Menschen *erfahren*, bezüglich dieses *Erfahrens* jedoch eine Unterscheidung zwischen dem Subjekt, das erfährt und dem Erfahrungsgegenstand selbst getroffen wird. Die Naturwissenschaftler untersuchen dabei den Erfahrungsgegenstand selbst, indem sie ihn einerseits erfahren aber gleichzeitig von sich als erfahrende Subjekte abstrahieren. Die Psychologie hingegen müsse laut Wundt sowohl den Menschen, der erfährt, als auch den Erfahrungsgegenstand selbst untersuchen ([Wu04], S. 1-3). „Den Zusammenhang der Erfahrungsinhalte, wie er dem Subjekt wirklich gegeben ist, kann nun die Psychologie nur aufzeigen, indem sie sich ihrerseits jener Abstraktionen und hypothetischen Hilfsbegriffe der Naturwissenschaft gänzlich enthält. Sind also Naturwissenschaft und

Psychologie beide in dem Sinne empirische Wissenschaften, dass sie die Erklärung der Erfahrung zu ihrem Inhalte haben, die sie nur von verschiedenen Standpunkten aus unternehmen, so ist doch die Psychologie in Folge der Eigentümlichkeit ihrer Aufgabe offenbar die strenger empirische Wissenschaft“ ([Wu04], S. 6). „In dieser Wundtschen Sichtweise [...] kann hier der Psychologie die Aufgabe gestellt werden, jene *allgemeingültigen Gesetzmäßigkeiten* herauszuarbeiten, nach denen sich die reale Welt als subjektive Erfahrung aufbaut“ ([Ho84], S. 21-22).

Zu Beginn des 20. Jahrhunderts entwickelte sich die Psychologie jedoch genau in die entgegengesetzte Richtung. Eine den Naturwissenschaften nachempfundene Psychologie setzte den Menschen als Untersuchungsgegenstand in den gleichen mittelbaren Kontext wie andere Erfahrungsgegenstände und verlangte (um wissenschaftlich zu sein), dass bei der Analyse dieses Untersuchungsgegenstandes (also des Menschen) genauso vom erfahrenden Subjekt zu abstrahieren sei wie bei den anderen Erfahrungsgegenständen auch. Um vom erfahrenden Subjekt zu abstrahieren und gleichzeitig das erfahrende Subjekt zu analysieren, musste eine *Außensicht* angenommen werden, die die menschliche Welt- und Selbsterfahrung ausklammert und das erfahrende Subjekt auf eine „Innenwelt“ reduziert, die „in der *empirisch unzugänglichen* »black box« zwischen *Reiz- und Reaktionsvariablen* [verschwindet]“ ([Ho84], S. 24).

Diese unterschiedlichen Ansätze implizieren ein unterschiedliches Konzept wissenschaftlicher Verallgemeinerung. Die ursprünglich angestrebte „*Überschreitung der Subjektivität auf die in ihr liegende intersubjektive Struktur*“ muß für die Variablenpsychologie, da sie das Subjektive ja in einen unaufhebbaren abstrakten Gegensatz zum Objektiven gebracht und aus der psychologischen Begrifflichkeit ausgeschlossen hat, *total unsichtbar*, ja *genuin unbegreiflich* bleiben.“ Zurückbleiben voneinander isolierte, austauschbare Subjekte. „Es ist nun fast naheliegend, die – nach der Ausklammerung ihres subjektiven Realitätsbezuges – verbleibenden isolierten Individuen durch Abstraktion von bestimmten Merkmalsverschiedenheiten als *gleichartige, voneinander unabhängige Elemente einer statistischen Verteilung*, also nach Art etwa einer Bohnen-Population, zu definieren. Damit sind dann die Anwendungsvoraussetzungen erfüllt für die statistische Schätzung von Stichproben auf Populationen“ ([Ho84], S. 25-26).

Aus diesem Grundverständnis entwickelte sich in der Psychologie der Behaviorismus. Dieser verstand sich nicht nur als wissenschaftstheoretischer Standpunkt, sondern insbesondere als Lerntheorie, die über fünfzig Jahre maßgeblich geprägt hat, was unter dem Begriff Lernen verstanden wird und wie daraus Lehren und Unterricht abgeleitet wird. Die behavioristische SR-(Stimulus-Response-) Psychologie verstand Lernprozesse als durch entsprechende äußere Umstände initiierte Verhaltensweisen. Der Kognitivismus wird generell als Überwindung des Behaviorismus betrachtet, da er die *empirisch unzugängliche black box* aufzubrechen scheint, indem er sich mit den „inneren“ Prozessen beim Lernen beschäftigt und Lernen als Informationsverarbeitungsprozess im Sinne eines informatischen Systems auffasst. Tatsächlich überwinden auch der Kognitivismus und die sich daran anschließenden Lerntheorien dieses Grundverständnis nicht, weil auch hier wieder das Verständnis einer

Psychologie als Wissenschaft über den Menschen vom Außenstandpunkt zugrunde liegt ([Ho95], S. 118-152). Erst der Konstruktivismus und Handlungstheorien gelangen zu einem subjektfokussierten Verständnis von Lernen und Handeln und machen Debatten über selbstorganisiertes, selbstbestimmtes Lernen wissenschaftlich möglich (siehe auch [GSIS], S. 13ff.).

3.2 Der Lehrlernkurzschluss

Auf Grundlage des behavioristischen Außenstandpunkts der gängigen Lerntheorien werden Lernprozesse durch Verstärkung, Instruktion, Lernziel-Vorgaben – also Lehren/Unterrichten von dritter Seite initiiert. Zu verstehen, dass intentionales, d.h. absichtliches und geplantes Lernen jedoch nur dann zustande kommt, wenn das erfahrende Subjekt selbst entsprechende Gründe für eine Lernhandlung hat ist daher erst möglich wenn der Außenstandpunkt aufgehoben wird.

Menschliches Lernen ist zwar eine auf Lernbedingungen bezogene, aber dennoch diesen gegenüber als eigenständige Aktivität konzeptualisierbar, die sich als begründetes Handeln aus den Lebensinteressen des Subjekts expliziert. Zum Lernen kommt es immer dann, wenn das Subjekt in seinem normalen Handlungsvollzug auf Hindernisse oder Widerstände gestoßen ist und sich dabei vor einer *Handlungsproblematik* sieht, die es nicht mit den aktuell verfügbaren Mitteln und Fähigkeiten, sondern nur durch den Zwischenschritt einer „Lernschleife“ überwinden kann. Es liegt im Interesse des Subjekts seine *Handlungsproblematik* (vorübergehend) als „Lernproblematik“ zu übernehmen, von der aus es seine weiteren Handlungen als spezifische Lernhandlungen strukturieren kann. Durch Lernen erweitert das erfahrende Subjekt seine Handlungsmöglichkeiten. Ein solches Handeln bezeichnet Holzkamp als *expansives Lernen* ([Ho95], 190 ff.).

Außengesetzte Lernbedingungen und -anforderungen, die mit den Lebens- und Lerninteressen des Subjekts nicht konform sind führen dazu, dass das Subjekt auf diese widerständig oder ausweichend reagiert und die erwarteten Tätigkeiten oder Lernresultate vortäuscht oder soweit wie erforderlich ausführt, um Sanktionen abzuwenden. Ein solches Handeln bezeichnet Holzkamp als *defensives Lernen*.

Die Vorstellung, man könne etwa durch Lehrpläne, Kompetenzmodelle oder didaktische Lehrstrategien Lernprozesse eindeutig vorausplanen, also Bedingungen herstellen, unter denen den Betroffenen nichts anderes übrigbleibt, als in der gewünschten Weise zu lernen, stellt einen *Lehrlernkurzschluss* dar, da erneut eine Außensicht auf SuS angenommen wird bei dem das Subjekt der Lernprozesse nicht eigentlich die Lernenden sind, sondern vielmehr die Lehrperson oder Lehrinstitution ([Ho95], 385 ff.). Ist die grundlegende Perspektive aus der das erfahrende Subjekt betrachtet wird, nicht geklärt, so kommt es erneut zu den bereits beschriebenen Verkürzungen, die die Debatte um selbstbestimmtes Lernen gerade überwinden will.

4 Lernen im Informatikunterricht

Die Datenanalyse im Forschungsprojekt *Computernutzungsbiographien* zeigt, dass SuS lange vor dem IU sowohl informatische als auch Computernutzungskompetenzen erwerben und zusätzlich ganz eigene, vielfach selbstorganisierte Lernformen entwickeln. Dabei erwarten die SuS vom IU, dass sowohl die Unterrichtsinhalte als auch deren Umsetzung am individuellen Standpunkt jedes einzelnen ansetzen und ihn entsprechend weiter fördern und fordern. Aus den Forschungsbereichen Didaktik und pädagogische Psychologie kommt die Forderung nach schülerzentriertem, handlungsorientiertem Unterricht, der auf einer konstruktivistischen Didaktik aufbaut.

Kann die Informatik, die anders als etablierte Fächer, bereits so manche Herausforderung zu meistern, den Forderungen nachkommen und den Wandel im Unterricht mitgestalten? Paradigmenwechsel sind ja deshalb so schwierig umzusetzen, weil hart erworbenes Wissen, Erfahrung und Fachkompetenz scheinbar aufgegeben werden müssen, zugunsten etwas Neues und damit auch Fremdes, das wieder neu erlernt und umsetzbar gemacht werden muss. Von der sich ständig weiter entwickelnden Fachwissenschaft Informatik sind wir es jedoch gewohnt mit der Dynamik der steten Veränderung umzugehen. Trotz unsicherer Stellung innerhalb der anderen Fächer, so manches mal belächelt oder als „überflüssig“ erklärt, kann das Schulfach Informatik vielleicht gerade hier sein Potenzial nutzen und diesen *didaktischen Wandel* aktiv mitgestalten.

Die *Bildungsstandards für die Informatik für die Sekundarstufe I* unterscheiden durchgehend zwischen Inhalts- und Prozessbereichen [GSIS]. Hier wird die Prozess- und Produktorientierung der Informatik deutlich und welches Potential sie für handlungsorientierten Unterricht und eine konstruktivistische Didaktik bietet. Die von Romeike dargestellten Kriterien kreativen Informatikunterrichts beinhalten unter anderem Offenheit in der Aufgabenstellung, dem Lösungsweg und der Ergebniserwartung, Relevanz und Identifikation für das lernende Subjekt sowie ein Unterrichtsklima der Vielfalt ([Ro08a], S. 82-85). Diese genannten Kriterien können direkt als Aspekte selbstorganisierten/selbstbestimmten Lernens abgeleitet werden und zeigen so weitere Überschneidungen dieses Bereichs mit Informatik. Förderung von Kreativität im IU impliziert so Formen des selbstorganisierten Lernens [Ro08a].

Selbstorganisiertes Lernen wird bereits vielfach als Unterrichtsmethode im Rahmen von Projektarbeit im IU eingesetzt. Witten, Penon und Dietz zeigen verschiedene konkrete Methoden für die Umsetzung von selbstorganisiertem Lernen im IU durch Kartenmethoden, Advance Organizer und Gruppenpuzzle dar [WPD06]. Darüber hinaus bietet der Berliner Bildungsserver im Bereich Lernfelder eine Plattform für selbstorganisiertes Lernen im IU an [SOL]. Hier sind bereits furchtbare Ansätze gemacht und die Informatikdidaktik in Forschung und Praxis kann hier weiter anknüpfen.

Lehrerinnen und Lehrer sind nicht dazu verdammt, sich an einem imaginären „Durchschnittsschüler“ zu orientieren und für alle SuS den gleichen Unterricht zu entwickeln. Die SuS kommen nicht nur mit unterschiedlichem Vorwissen und Erwartungen in den IU, sondern auch mit unterschiedlichen Fähigkeiten und Methoden zu lernen. Gerade dieses unterschiedliche Potenzial können LuL und SuS gemeinsam nutzen, um einen IU der Vielfalt zu gestalten.

Literaturverzeichnis

- [Ba02] Bannach, M.: Selbstbestimmtes Lernen. Freie Arbeit an selbst gewählten Themen. Schneider-Verl. Hohengehren, Baltmannsweiler, 2002.
- [GSIS] Grundsätze und Standards für die Informatik in der Schule - Bildungsstandards Informatik für die Sekundarstufe I. In (Arbeitskreis „Bildungsstandards“, GI Hrgs.)LOG IN, 2008, 28.
- [BMPF] Bundesministerium für Bildung und Forschung: Zukunft Bildung und Betreuung (IZBB). www.ganztagsschulen.org.
- [Bö03] Böhm, A.: Theoretisches Codieren: Textanalyse in der Grounded Theory. In (Flick, U.; Kardorff, E. von; Steinke, I. Hrsg.): Qualitative Forschung. Ein Handbuch. Rowohlt, 2003; S. 457–485.
- [Fa99] Faulstich, P.: Einige Grundfragen zur Diskussion um "selbstgesteuertes Lernen". In DIE-Materialien, 1999, 18; S. 24–39.
- [FHJ03] Furtner-Kallmünzer, M.; et. al.: In der Freizeit für das Leben lernen. VS Verlag, 2003.
- [He85] Hentig, H. von: Die Menschen stärken, die Sachen klären. Ein Plädoyer für die Wiederherstellung der Aufklärung. Reclam, Stuttgart, 1985.
- [Ho84] Holzkamp, K.: Die Bedeutung der Freudschen Psychoanalyse für die marxistisch fundierte Psychologie. In Forum Kritische Psychologie, 1984, 13; S. 15–40.
- [Ho95] Holzkamp, K.: Lernen. Subjektwissenschaftliche Grundlegung. Campus Verlag, Frankfurt am Main, 1995.
- [Kn09] Knobelsdorf, M.: Students' Pathways to CS – A Biographical Research Approach. In ACM Transactions on Computing Education (TOCE), 2009, in Press.
- [KR08] Knobelsdorf, M.; Romeike, R.: Creativity as a pathway to computer science: ITiCSE '08: Proceedings of the 13th annual conference on Innovation and Technology in Computer Science Education. ACM, 2008; S. 286–290.
- [KS07] Knobelsdorf, M.; Schulte, C.: Das informatische Weltbild von Studierenden. In (Schubert, S. Hrsg.): Didaktik der Informatik in Theorie und Praxis. 12. GI-Fachtagung Informatik und Schule - INFOS 2007, Siegen, 2007; S. 69–79.

- [Ku07] Kuckartz et al.: MAXQDA. www.maxqda.de. VERBI Software. Consult. Sozialforschung. GmbH Germany, 2007.
- [Ma07] Mayring, P.: Qualitative Inhaltsanalyse. Grundlagen und Techniken. BELTZ Deutscher Studien Verlag, 2007.
- [RBS] Robert Bosch Stiftung: Der Deutsche Schulpreis. schulpreis.bosch-stiftung.de.
- [Ro08a] R. Romeike: Bildungsstandards kreativ erreichen - ein Unterrichtskonzept für die Sekundarstufe I. In (Brinda, T. et. al. Hrgs.): Didaktik der Informatik - Aktuelle Forschungsergebnisse. 5. Workshop der GI-Fachgruppe "Didaktik der Informatik", Erlangen. (LNI) 2008, 135.
- [Ro08b] Romeike, R.: Kreativität im Informatikunterricht. Dissertationsschrift. Universität Potsdam, 2008.
- [SOL] Penon, J.: Selbstorganisiertes Lernen (SOL). <http://www.bebis.de/themen/lernfelder/sol>.
- [Wi04] Wirth, J.: Selbstregulation von Lernprozessen. Waxmann, 2004.
- [Wo00] Wosnitza, M.: Motiviertes selbstgesteuertes Lernen im Studium. Verlag Empirische Pädagogik, 2000.
- [WPD06] Witten, H.; Penon, J.; Dietz, A.: SOL – Schule ohne Lehrer? Selbstorganisiertes Lernen im Informatikunterricht. In LOG IN, 2006, 26; S. 74–81.
- [Wu04] Wundt, W.: Grundriss der Psychologie. VDM Müller, Düsseldorf, 2004.
- [WWE03] Witthaus, U.; Wittwer, W.; Espe, C.: Selbst gesteuertes Lernen. Theoretische und praktische Zugänge, Bertelsmann, Bielefeld, 2003.

Integration statt Sahnehäubchen

Die technologische Basis der Kulturtechniken hat sich verändert

Dr. Elin-Birgit Berndt
Fachbereich 3 AG Digitale Medien in der Bildung
Universität Bremen
Bibliotheksstraße
28334 Bremen
berndt@uni-bremen.de

Abstract: Das Scheitern der Integration der IuKT in den Unterricht resultiert aus falschen Erwartungen an den Einsatz digitaler Medien im Unterricht. Nicht der „Mehrwert“ taugt als Kriterium. Die Werkzeugmetapher verstellt den Blick auf den allgemeinbildenden Charakter informatischer Bildung. Die Veränderung der technologischen Basis der Kulturtechniken ist der Schlüssel: Digitale Medien sind Instrumente der Weltaneignung, ihrer Erschließung und Erkenntnis.

„Lernen Kinder wirklich durch digitale Medien?“ Unter dieser Fragestellung erschien vorletztes Jahr in der Süddeutschen Zeitung ein Artikel mit dem Titel „Der Laptop-Flop“. Der Autor berichtete:

*„In den USA mehren sich Zweifel daran. Einige Schulen haben Computer wieder aus den Klassenzimmern verbannt. Mark Warschauer hat es momentan nicht gerade einfach. Eigentlich ist es der Pädagogik-Professor an der University of California in Irvine nicht gewohnt, dass ihm ein derart scharfer Wind ins Gesicht bläst. Die Arbeit mit Computern fördere die Kreativität und Selbstständigkeit von Jugendlichen, hatte Warschauer immer wieder gepredigt und das in seinem 2006 erschienenen Buch *Laptops and Literacy: Learning in the Wireless Classroom* auch niedergeschrieben.*

Bisher fanden das alle immer gut. Allen voran die amerikanische Regierung, die in den vergangenen Jahren Milliarden in die flächendeckende Ausrüstung von Schulen mit Computern und Laptops gesteckt hatte. Seit einiger Zeit jedoch ist ausgerechnet im Land der Bits und Bytes eine Kehrtwende in Sachen digitale Erziehung zu beobachten. Manche Schulen nehmen den Teenagern die Laptops wieder weg – auch weil die Schüler damit lieber auf Seiten pornografischen Inhalts im Internet surfen, statt ihre Hausaufgaben zu erledigen.“ [Ju 07]

Eine wissenschaftliche Antwort auf seine eingangs gestellte Frage gäbe es bis heute nicht: „Auch nach sieben Jahren haben wir keinen Beleg dafür, dass der Einsatz von Computern im Unterricht die Leistung der Schüler verbessert hätte“, zitierte er die New York Times.

1 Nicht mit dem Medium, sondern im Digitalen Medium wird gelernt

„Lernen Kinder wirklich durch digitale Medien?“ – das ist die falsche Frage. Ob digitale Medien in Schulen eingesetzt werden sollen, hängt nicht davon ab, ob mit ihnen besser

gelernt wird. Der Einsatz ist geboten, weil wir uns **in** digitalen Medien bewegen, in ihnen lernen – nicht nur mit oder durch digitale Medien. Basis der Techniken der Wahrnehmung, der Aneignung und der Gestaltung von Welt – Basis der so genannten „Kulturtechniken“ – ist die Digitalität. Die Realität, in der wir uns bewegen, erfahren wir zwar über unsere Sinne, aber wir bedienen uns dabei schon lange, nicht erst seit den letzten beiden Jahrhunderten technischer Mittel, die unsere Sinne und Organe erweitern und „verlängern“, die über die Dimensionen unseres Auges, unseres Ohres hinausgehen. Und auch vormals unseren Gehirnen überlassene Tätigkeiten wurden „maschinisiert“. Digitale Medien be- und verarbeiten Daten, speichern oder übertragen sie nicht nur. Das muss der Ausgangspunkt schulischen Einsatzes der IuKT sein. Diese Kritik wurde schon vor 16 Jahren von mir auf der Tagung der GI zu Computer und Schule vorgetragen. [vgl. BI93]

Dass die technologische Basis der Kulturtechniken Schreiben, Rechnen, Lesen [Be07,123] vgl. [Wa04] sich im digitalen Medium radikal verändert hat, wird bisher aber nicht zum Ausgangspunkt der Integration in Schulen gemacht. Eher geht es darum, dass die jugendlichen Anwender fit fürs Büro, für den Beruf gemacht werden. Die Technik soll möglichst wenig auffallen. Der Blick wird eingeschränkt auf einen störungsfreien Arbeitsprozess. Der Stellenwert technischen Wissens und technischen Verständnisses für die Allgemeinbildung, für Bildungsprozesse überhaupt, wird gering geschätzt, nur Fertigkeiten, die Handhabung von „Werkzeugen“ zählen – „Ein bisschen Word“! [Ha09]- weil sie für einen reibungslosen Ablauf in Produktionsprozessen vorteilhaft sind. Die Bezeichnung der Software als „Werkzeug“ belässt den Umgang mit dem Computer in der Tradition handwerklichen Arbeitens, in der die virtuose Handhabung eines Instruments das Ziel ist, in der Erfahrung, Übung und Geschick zählen. Das Wissen bleibt implizites Wissen, der „Novize“ ahmt den „Meister“ nach. Es ist die Werkzeug-Metapher, die den Blick trübt und die Sicht auf die informationstechnischen Abläufe hinter der Oberfläche verstellt. [Be07,123]

So werden in der Schule Digitale Medien den Kindern und Jugendlichen als

- „Werkzeuge“ dargeboten, die ihnen behilflich sein sollen bei der Erledigung schulischer Arbeiten wie z. B. Hausaufgaben, Referate oder Recherche.
- oder als „Lernwerkzeug“ in Gestalt von Lernsoftware.

Wenn wir uns nicht bewusst werden, dass wir uns im Digitalen Medium in der Welt bewegen, dass wir uns die Zugänge zur Welt im Digitalen Medium bahnen, dass wir uns Digitale Medien aneignen, um Erkenntnisse und Einsichten zu erlangen, dann bleibt uns der „Wolf im Schafspelz“ verborgen, dann begreifen Schüler nicht, dass hinter den Icons, auf die sie klicken, sich „Maschinen“ verstecken, mit denen sie interagieren. Moderne Benutzer-Oberflächen täuschen darüber hinweg, dass wir nur einem Rechner Befehle erteilen, wenn wir meinen, am PC zu schreiben, zu lesen, zu gestalten oder zu recherchieren.

Die Metapher trübt den Blick und verstellt die Sicht auf die informationstechnischen Abläufe hinter der Oberfläche.

Besteht doch die wesentliche Differenz zu den „alten“ Medien darin, dass digitale Medien Daten/Zeichen verarbeiten, statt sie (nur) zu vermitteln oder zu speichern. Deshalb bieten sie sich an zur Selbstreflexivision, zur Dokumentation von Aneignungsprozessen (student records) und zu metakognitiven Diskursen.

2 Integration in das Schulfach: z. B. Deutsch

Obwohl es z. B. gerade für den Deutschunterricht (DU) schon seit Anfang der 90er Jahre sehr viele Vorschläge für den Einsatz von Computern im DU gibt, die durch etliche Konzepte und Modellversuche wie z. B. „SEMIK“ in den folgenden Jahren ergänzt wurden, sind digitale Medien immer noch nicht in den an den Schulen praktizierten Deutschunterricht integriert. Die fachdidaktische Literatur dokumentiert in Sammelbänden und Fachzeitschriften die didaktische Forschung und den Diskurs zu einem kreativen, mediengestützten Deutschunterrichts, auf den großen Tagungen der Fachverbände bleibt das Thema seit Mitte der 90er Jahre keineswegs ausgespart, so z. B. 2007 weder bei den Linguisten:

„...welche Kompetenzen spezifisch sind für Kinder und Jugendliche im Umgang mit Medien: Was ist anders an der Situation als an derjenigen von Erwachsenen? ... Das Verfassen von Bewerbungsschreiben muss ebenso neu gelernt werden wie das Ansprechen unbekannter KommunikationspartnerInnen im Chatroom. Zusätzlich müssen Kinder und Jugendliche Kompetenzen in Transferleistungen entwickeln, in denen sie im Allgemeinen nur wenig geschult werden. Groß- und Kleinschreibung kann z. B. in Chats so verwendet werden, wie gelernt, aber ebenso davon abweichend; Bildbeschreibungen in ... Blogs können ähnlich denjenigen in der Schule verfasst werden, aber auch in anderen Stil- und Registervarianten; Texte auf Homepages können sich an gelernten Erzählmustern orientieren, aber genauso z.B. rein assoziativ organisiert sein. Welche Aspekte transferiert werden und welche nicht, nach welchen Kriterien Kinder und Jugendliche entscheiden, bestimmte Elemente zu verwenden, zu verwerfen, zu ändern bzw. anders zu funktionalisieren, sind einige der offenen Fragen, ... [GAL07]

noch beim Germanistenverband:

„Die Beschäftigung mit Medialität von Sprache, Literatur und Kunst setzt sich zunächst mit den Bedingungen und Folgen der fundamentalen These auseinander, dass alle drei immer (nur) in sinnlich wahrnehmbarer Vermittlung existieren, jedenfalls nur vermittelt durch Sinne zugänglich sind. Medialität ist deshalb schon konstitutiv für die leibgebundenen Formen der direkten Kommunikation, mehr noch für die technisch und institutionell geprägten Formen indirekter, „medienvermittelter“ Kommunikation; beide haben deshalb immer auch eine materiale Seite. Wie wirken die Potenziale und Beschränkungen der Sinne, der physikalischen und schließlich der technischen Medien in Sprache, Literatur und Kunst? Wie befördern und erschweren sie das Zusammenspiel verschiedener Sinne und der unterschiedlichen Symbolsysteme, die sie ermöglichen? Wie verhält sich der Aspekt der Medialität und Materialität der Kommunikation zur Frage von Universalität und Kulturalität?“ Sektion 4: Medialität von Sprache, Literatur und Kunst [DG07]

Empirische Studien zum Computereinsatz in den Schulen [Ifib07] und. [Bu06] zeigen, dass weder die innerhalb der Fachcommunities erarbeiteten Konzepte noch die über Lehrerportale wie z. B. „lehrer-online“ verbreiteten Materialien oder die in verschie-

denen Foren wie z. B. „zum.de“ diskutierten Unterrichtsideen breit aufgegriffen werden, bestenfalls zögerlich. Als Begründungen werden dort vorrangig genannt.

- die nicht hinreichende Ausstattung der Schulen,
- die unflexible Unterrichtsorganisation
- und – immer noch – die nicht zureichende Fortbildung der Lehrkräfte.

Die aktuellen Studien zum Einsatz des Computers in allgemeinbildenden Schulen zeigen, dass die digitalen Medien in der Schule zum einen noch wenig zu einer Veränderung von Unterricht und Lernen beigetragen haben und zum anderen, dass über die neuen Medien selbst wenig gelernt werden soll. So stellte die Auditierung des Förderprogramms Neue Medien in der Bildung – Förderbereich Schule [Au04] zu „didaktischen Mehrwert“ und Innovationen Ende 2004 fest:

„Trotz des in vielen Produkten vorhandenen hohen Potenzials zur Unterstützung von Lehr- und Lernformen sowie zur Erprobung neuer Erfahrungsräume sind in den wenigsten Fällen die Möglichkeiten digitaler Medien ausgeschöpft. (...) Innovationen ... sind nicht erkennbar. Dies gilt auch für die mit den Produkten verbundenen Arbeitsformen. Hier werden noch in deutlich zu geringem Maße kooperative und kollaborative telemediale Arbeitsformen berücksichtigt. Phasen der Lernreflexion fehlen fast völlig.“ [Au04,75]

Anlässlich des 10jährigen Jubiläums von „Schulen ans Netz“ präsentierten Bardo Herzig und Silke Gräfe im Oktober 2006 in Berlin eine aktuelle Übersichtsstudie vgl. [HG06] zum Lernen mit digitalen Medien in deutschen Schulen und leiteten Handlungsnotwendigkeiten für die am Bildungsprozess Beteiligten ab. Sie wiesen insbesondere auf die Schere zwischen häuslicher und schulischer Nutzung und somit zwischen formellem und informellem Lernen hin.

„In Bezug auf die Nutzung von Computer und Internet im Unterricht kann insgesamt auch noch von keiner selbstverständlichen Integration digitaler Medien in den Unterricht gesprochen werden, wenngleich auch hier die Nutzungsdaten in den letzten Jahren zunehmen. In den Grundschulen erfolgt ein häufiger Computereinsatz vor allem in den Fächern Deutsch und Mathematik sowie in Arbeitsgemeinschaften, wobei vorrangig Lernsoftware und multimediale Nachschlagewerke zum Einsatz kommen. In den Sekundarstufen I und II ist ein häufiger Einsatz digitaler Medien vor allem im Fach Informatik und in Arbeitsgemeinschaften zu finden, gefolgt von technischen, naturwissenschaftlichen und wirtschaftlichen Fächern. An Software werden vor allem Werkzeuge, Präsentationsprogramme und Programmiersprachen verwendet. (...) In der Tendenz zeigen die empirischen Daten, dass je nach Schulart eine Kerngruppe im Umfang von 10 bis 30% der Lehrpersonen zu den regelmäßigen Anwendern digitaler Medien im Unterricht gehört. Solche Gruppen stellen auch die hauptsächlichsten Nutzer von Online-Angeboten für Lehrer dar.“ [HG06,14] (...) „Insgesamt wird erkennbar, dass im Hinblick auf die Unterrichts- und Lernkultur Veränderungen mit dem Einsatz digitaler Medien einhergehen; ob dies allerdings zu diesem Zeitpunkt bereits als eine neue Lernkultur bezeichnet werden kann, ist fraglich. Vielfach sind Veränderungen auf den Einbezug der Medien fokussiert, legen allerdings auch dort bereits ein modifiziertes Rollenverständnis nahe. Eine Änderung der grundlegenden didaktischen Muster zur

Integration digitaler Medien in den Unterricht, die auch in neuen Formen der fachlichen Auseinandersetzung mit Inhalten die Potenziale digitaler Medien zur Entfaltung bringt, erfordert jedoch noch deutlich weiter gehende Anstrengungen.“ [HG06,17]

Die KIM- und JIM-Studien des letzten Jahres weisen ebenfalls deutlich die Diskrepanz zwischen häuslicher Nutzung und schulischer Nutzung nach.

Eine Bestandsaufnahme im Auftrag des Bundesministeriums für Bildung und Forschung (BMBF) von 2008 mit dem Titel

„Digitale Schule – wie Lehrer Angebote im Internet nutzen“

kommt zu folgenden Ergebnissen hinsichtlich der Nutzung von Online-Lehrer-Angeboten durch Lehrer:

„1. Allgemeinbildende und berufsbildende Schulen bieten ein großes Potenzial für die Nutzung von Online-Medien, das noch lange nicht ausgeschöpft wird. Die erforderliche Ausstattung ist sowohl bei Lehrern als auch bei Schülern größtenteils vorhanden, aber die Nutzung könnte deutlich höher sein.

2. Die Nutzung von Online-Angeboten für Lehrer zur Unterrichtsvorbereitung ist für die meisten Lehrerinnen und Lehrer eine Selbstverständlichkeit geworden.

Die Nutzung von Online-Angeboten im Unterricht ist hingegen eher die Ausnahme.

3. Es existiert ein großes und umfangreiches Online-Angebot für Lehrer. Die vorliegende Untersuchung ermittelt 36 schulspezifische Websites, die von Lehrern häufig genutzt werden. Hinzu kommen Angebote, die sich auch an andere Zielgruppen wenden.

4. Es kristallisieren sich vier Angebots-„Typen“ heraus: 1. Kostenlose Informationsangebote mit Verweisen auf Inhalte anderer Anbieter; 2. Bereitstellung von Unterrichtsmaterialien speziell für Lehrer (z.T. kostenlos); 3. Spezialangebote für bestimmte Lehrer-Zielgruppen oder Themen; 4. Arbeitsumgebungen für die Zusammenarbeit von Lehrern, Schülern und anderen schulnahen Akteuren.

5. Bei den Angeboten und Funktionalitäten legen Lehrer vor allem Wert auf Datenbanken mit Lerninhalten sowie auf Links und aktuelle Informationen für Lehrer. Geringer ist die Präferenz für virtuelle Arbeitsumgebungen.

6. Hauptaufgabe einer Portalredaktion ist nach Ansicht der Lehrer die Erstellung von Contents im eigenen Hause. Experten (Schuldirektoren und Vertreter der Länder) befürworten hingegen den Aufbau eines Netzwerks von Lehrern, die die Inhalte für das Portal erstellen.

7. Eine Erweiterung der Zielgruppen von Web-Angeboten für Lehrer wird eher nicht gewünscht, wohl aber eine gezielte Öffnung für Referendare und Pädagogik-Studierende, also den Lehrer-Nachwuchs.

8. Weitgehend akzeptiert werden von Lehrern „klassische“ Online-Angebote im Sinne des „Web 1.0“, also von professionellen Autoren erstellte Texte und Grafiken, die keine weitere Bearbeitung mehr ermöglichen oder benötigen. Eine kleine Pioniergruppe von Lehrern und Experten begrüßt die neuen Formen des „Web 2.0“, in denen alle Nutzer an den Inhalten im Web mitarbeiten. ...“

Die Verfasser der Studie stellen weiter fest:

„Während Computer in Haushalten mit Kindern zum Alltag gehören und als Spielkonsole, Hausaufgabenhelfer und Chat-Kanal genutzt werden, sind neue Lernmedien in den Schulen bisher nur bedingt angekommen. Zwar bereiten die meisten

Lehrer ihren Unterricht auf ihrem heimischen Computer vor, doch im Unterricht selbst werden Internet und CD-Rom nur selten eingesetzt.“ [Bu09]

MediaPerspektiven 12/2008, und die JIM/KIM-Studien bestätigen diese Ergebnisse.

Die Zielgruppe der Lehrer sei inzwischen in der digitalen Welt angekommen: „Nachdem sie lange Zeit mit dem Computer und dem Internet eher zurückhaltend umgingen, nutzen mittlerweile 90 Prozent der Lehrer Computer und Internet zur Unterrichtsvorbereitung; damit ist das Internet ein wichtiges Werkzeug der Lehrer. Im Unterricht selbst nutzen Lehrer die neuen Medien noch immer sehr selten. Lediglich 20 bis 30 Prozent der Lehrer setzen Computer und Internet zu Unterrichtszwecken ein.

Aus Sicht der Schüler fristet damit ein Medium nur eine Randexistenz, das für sie zu Hause längst eine Selbstverständlichkeit ist: 83 Prozent der Kinder zwischen 6 und 13 Jahren und 98 Prozent der Jugendlichen haben zu Hause Zugriff auf PC und Internet.“ Allerdings sei die Akzeptanz auch abhängig von den vorhandenen Erfahrungen. Lehrer, die Computer seltener nutzen, hätten eine skeptischere Einstellung.

Von einer grundlegenden Änderung der Lernkultur seien die Schulen noch weit entfernt, so das Fazit der Studie. Wenn auch aus dem Alltag von Lehrerinnen und Lehrern Computer und Internet nicht mehr wegzudenken seien, müsse aber festgehalten werden, dass Lehrer das Internet überwiegend zur Unterrichtsvorbereitung nutzen würden. Im Unterricht selbst würden die Neuen Medien eher selten eingesetzt, so die Ergebnisse der Studie. [vgl. Bu09]

Die mangelnde Akzeptanz und geringe Attraktivität der digitalen Medien in schulischen Bildungsprozessen führe ich darauf zurück, dass der Computereinsatz auf die Nutzung als „Werkzeug“ zurechtgestutzt wird. Z. B wird in Schreibprozessen die Digitalität und die Medialität von Schrift nicht gebührend beachtet, wie ich in „Warum die Werkzeug-Metapher in Bildungsprozessen in die Irre führt“ [Be07]erörtert habe. Eine Bremer Studie bestärkt mich darin:

Die Ausstattung z. B. der Bremer Schulen und die Einstellungen von Lehrern und Schülern zum Computereinsatz werden seit längerem vom Bremer Institut für Informationsmanagement dokumentiert. Für 2006 liegen Studien zu Bremen und Bremerhaven vor. [vgl. Ifib07]

Aus dem Fazit der Autoren der Studie lässt sich entnehmen, dass bei relativ guter Ausstattung der Schulen die Nutzung des PC als Werkzeug dominiert.

Die Einstellung der Lehrer zum Computereinsatz wird wie folgt zusammengefasst:

Offensichtlich reduziert sich der Computereinsatz auf eine allgemeine Einführung in die Handhabung und auf die Standardsoftware und ihre instrumentelle Nutzung, spezifische Konzepte für einen medienintegrierten Deutschunterricht sind der Studie nicht zu entnehmen. Das bestätigen auch die Befragungen der Schüler hinsichtlich der unterrichtlichen Nutzung der Computer. Demgegenüber stehen die Wünsche der Schüler an die schulische Einbeziehung von Computern: Schüler wünschen sich eine stärkere Integration in den Unterricht. Sie beklagen, dass der Computer im Fachunterricht keine Rolle spielt. Die Beobachtungen an Bremer Schulen werden durch die jüngsten JIM und KIM Studien bestätigt.

Die Autoren der Bremer Studie fassen die Ergebnisse ihrer Befragungen dahingehend zusammen:

„Dabei wird die Vermittlung von Medienkompetenz am häufigsten genannt, es folgen eher werkzeugorientierte Nutzungsformen zur Vermittlung von Basiskompetenzen (z.B.

Computerführerscheine, Office, Internet). Aussagen zur integrativen Nutzung im Fachunterricht werden meist nur punktuell von einigen Schulen gemacht. (...)

Die Schwerpunkte liegen dabei auf Arbeitsgemeinschaften, die den Computer selbst oder die Schulhomepage der Schule zum Gegenstand haben, sowie klassische AGs, die sich gut durch Medien unterstützen lassen (z.B. die Schülerzeitung). (...)" [Ifib 2006, 33 ff].

Aus den detaillierten Angaben zur Ausstattung der Schulen in der Studie lässt sich schließen, dass etliche der Angebote für den Deutschunterricht, wie sie in der fachdidaktischen Forschung diskutiert werden, technisch realisierbar wären, auch wenn die Bedingungen noch nicht so optimal sind wie z. B. in Laptop Schulen [vgl. SI02].

Wenn man vom „Werkzeug“ spricht, wird z. B. in Schreibprozessen die Medialität und die Digitalität von Schrift nicht gebührend beachtet, es bedarf eines Medialitätsbewusstseins: Weder in den „Bildungsstandards“ noch in neueren Schulbüchern wird der Veränderung der technologischen Basis der Kulturtechniken Rechnung getragen. In diesem Zusammenhang ist insbesondere darauf hinzuweisen, dass gerade auch in den Verlautbarungen der Kultusministerien (Bildungsstandards u.a.) und in aktuellen Schulbüchern nicht von einem grundlegenden Wandel z. B. der Schreib- und Lesetechnologie als Basis der in der Schule zu vermittelnden Kulturtechniken ausgegangen wird. [vgl. dazu VdS07] Nicht hinreichend wird reflektiert, dass die technologische Basis der Kulturtechniken Schreiben, Rechnen, Lesen sich im digitalen Medium radikal verändert hat. Wolf-Rüdiger Wagner hat Bildungsstandards daraufhin untersucht, wieweit dort der Aneignung von Medienkompetenz überhaupt ein Stellenwert eingeräumt wird:

„In den von der KMK veröffentlichten Bildungsstandards für Deutsch, die erste Fremdsprache und die Naturwissenschaften (Biologie, Physik und Chemie) wird nicht explizit auf Medienbildung Bezug genommen, es wird jedoch die Vermittlung einer Reihe von Kompetenzen gefordert, die Bestandteil von Medienbildung sind. ... Medienbezogene Kompetenzen finden sich in den Bildungsstandards zum einen in den Bereichen „Texte/Medien verstehen“, „Informationen suchen und bewerten“, „Kommunizieren und Präsentieren“, zum anderen dort, wo es in den Bildungsstandards darum geht, mit Medien zu arbeiten, zu lernen, Erkenntnisse zu gewinnen und über die Grenzen der Medien als Werkzeuge zur Wissensaneignung zu reflektieren“. [Wa08,34f] Deutlich wird, dass nicht von einem grundlegenden Wandel der Schreib- und Lesetechnologie als Basis für alle im DU zu verrichtenden Tätigkeiten ausgegangen wird, sondern dass am „Handschriften“ festgehalten wird – bestenfalls unter Zuhilfenahme von Print-Lexika oder Internet-Recherchen. Klausuren am PC werden nicht thematisiert. Digitale Jahres-/Hausarbeiten sind Ausnahmen.

Weder wird gebührend beachtet, dass die Menschen die Reichweite ihrer Sinnesorgane längst durch technische „Erweiterungen“ verlassen haben, noch wird der Entwicklung eines dem entsprechenden Medialitätsbewusstseins Raum gegeben. Wagner führt diese Ignoranz auf ein fehlendes Medialitätsbewusstsein zurück. „Aus der Perspektive des Medialitätsbewusstseins werden Medien wichtig, weil sie die organischen Möglichkeiten des Menschen zur Kommunikation und Wahrnehmung, einschließlich der damit verbundenen Prozesse der Informationserfassung, -verarbeitung, -speicherung und -übermittlung, ersetzen, entlasten oder überbieten.“ [Wa08] Er hebt hervor – wie auch schon in *Medienkompetenz revisited* [Wa04], dass versäumt worden sei, die Organmetapher in Bildungsprozessen zu thematisieren:

„Mit dem Bezug auf die organischen Möglichkeiten der Menschen wird der Blick auf die spezifischen Leistungen der Medien für die Ausdehnung und Erweiterung unseres Kommunikations- und Erfahrungshorizonts gelenkt. Schon Alexander von Humboldt benutzte diese „Organmetapher“, um deutlich zu machen, dass die Entwicklung "neuer Organe" für das neuzeitliche Denken insgesamt grundlegend war (...)

Die von Humboldt vorgenommene Ausweitung des Medienbegriffs ist aus heutiger Sicht nicht erstaunlich, denn Thermometer und Barometer sind nichts anderes als Vorläufer der diversen Sensoren, Detektoren und Messfühler, die heute als Peripheriegeräte den Computer zum Universalmedium machen.

Mit der Organmetapher arbeitete auch Vannevar Bush, als er in seinem Essay „As we may think“ das Konzept des Memex (Memory Extender) entwickelte, der als ein Vorläufer des Personal Computers und des Hypertextes gilt. Nach Jahrhunderten, in denen die sensorischen Fähigkeiten der Menschen durch immer perfektere Techniken zur Informationserfassung und -speicherung gesteigert worden sind, käme es darauf an Techniken zur Verstärkung der menschlichen Fähigkeiten zur Informationsverarbeitung einzusetzen.“ [Wa08,34 ff]

Wolf-Rüdiger Wagner hat in seinen Publikationen immer wieder darauf hingewiesen, dass ein Medialitätsbewusstsein unverzichtbar sei, so postuliert er:

„Da Medien in allen Bereichen unverzichtbare Werkzeuge der Weltaneignung geworden sind, bietet es sich an, die Forderung nach Vermittlung von Medialitätsbewusstsein weiter zu fassen; denn Medien konstruieren, inszenieren und interpretieren „Wirklichkeit“ in einer jeweils medienspezifischen Art und Weise. Medien verdoppeln nicht die „Wirklichkeit“, sondern zeigen bestimmte Ausschnitte und ermöglichen den Zugang zu bestimmten Aspekten der „Wirklichkeit“. [Wa08,34 ff]

Dabei ist für Wagner entscheidend, dass „die zunehmende Medialisierung der Gesellschaft (...) – zumindest in der Wissenschaft – zu einem wachsenden Bewusstsein von der Konstruiertheit der Wirklichkeit geführt“ habe. Prinzipiell sei eine solche Konstruiertheit natürlich auch bereits für die Fotografie anzusetzen, mit der gerade keine Realitätsabbildung geleistet werde. „Während der Prozess der Wirklichkeitskonstruktion bei der Fotografie auf den ersten Blick häufig verborgen bleibe, wird er heutzutage beim Fernsehen deutlicher sichtbar – z. B. bei der Live-Übertragung eines Fußballspiels, wo mehrere Kameras in unterschiedlichen Positionen zum Einsatz kommen und Superzeitlupe als ein Analyseinstrument zur Verfügung steht, wie es dem Menschen (bzw. hier: dem Schiedsrichter) allein von seinen Sinnen her nicht gegeben ist.“ konstatiert Wagner.

3 Die Angst vor dem „Verlust“ dominiert

Den bildungspolitischen Konzepten und den Studien zur Computernutzung möchte ich die Argumente von Lehramtsstudierenden an der Bremer Uni gegenüberstellen, die mir in meinen Lehrveranstaltungen zum „Lernen mit technischen Medien“ immer wieder in den letzten 5 Jahren entgegengehalten worden sind, wenn es um die Integration des Computers in den Deutschunterricht ging. Meine Zitate stammen aus nicht veröffentlichten Evaluationen, aus Chat- und Forenbeiträgen in meinen Lehrveranstaltungen der letzten beiden Jahre. Die Einstellung der Studierenden wurde nicht systematisch erhoben, die Teilnehmer der Lehrveranstaltungen können auch nicht

als repräsentative Population für Bremer Lehrämter gesehen werden. Dennoch halte ich aufgrund meiner Erfahrungen in diesen Lehrveranstaltungen die von den Studierenden formulierten Bedenken für aufschlussreich. Sie stützen meine Thesen zum Scheitern einer Integration Digitaler Medien in das Schulfach Deutsch.

Wie wenig ein Medialitätsbewusstsein auch bei Lehramtsstudierenden vorausgesetzt werden kann, hat sich immer wieder in heftigen Diskussionen gezeigt, die oft darauf hinausliefen, dass ihnen der Verlust der Handschrift am bedrohlichsten zu sein scheint. Einem integrierten Einsatz digitaler Medien im Deutschunterricht steht die Befürchtung vieler Lehramtsstudenten-Studenten entgegen, dass dann die Schüler nicht mehr flüssig, sicher und schnell mit der Hand schreiben können. Sie beobachten an sich selbst, dass das Schreiben mit der Hand, wenn es nicht mehr praktiziert wird, verlernt wird, wie andere „Handfertigkeiten“ auch: häkeln, stopfen, stricken, usf. Sie spüren diesen Verlust einer Fertigkeit bereits an sich selbst im Studium, manche befürchten z. B. für die handschriftlich auszufertigenden Klausuren nicht mehr fit genug zu sein. Diese Sorge erschwert die Akzeptanz der digitalen Medien in der Schule, insbesondere in Schreibprozessen.

Gefragt, was sie derzeit hindere, als Lehrer den Computer in Schreibprozessen einzusetzen und welche ihrer Bedenken gegen Schreibprozesse am Computer nicht ausgeräumt werden konnten, debattierten die Studierenden in Foren, Chats und Evaluationsfragebögen in meinen Lehrveranstaltungen. (Diese Daten sind bisher unveröffentlicht. Die Zitate werden unredigiert wiedergegeben, die Namen sind abgeändert.) Wie die Studenten ihre Beobachtungen bewerten und inwiefern diese deren Akzeptanz der digitalen Medien in der Schule, insbesondere in Schreibprozessen mindern, soll an einigen mir besonders aufschlussreichen Zitaten aus dem Juni 2007 sichtbar gemacht werden:

„Ich denke nur einfach, dass das Handschriftliche nicht vernachlässigt werden darf. man wird so faul. Und die ganzen Abkürzungen tragen dazu bei, dass die deutsche Sprache verkommt. (...) bestes Beispiel das Chatten und die SMS. Und Schüler sollten sich einfach die Mühe machen, ein Referat niederzuschreiben, mit der Hand. Computer können ja meinetwegen eingesetzt werden ... aber in Maßen!“ (Verena)

„Eine gescheite Schriftlichkeit lässt sich nur per Hand und nicht mit dem PC herstellen.“ (Arnd)

„(...) dass der Erwerb von graphomotorischen Fähigkeiten nicht zu kurz kommen darf. Deshalb müssen und dürfen meines Erachtens Textverarbeitungsprogramme aber nicht aus Grundschulklassen verbannt werden. (...) Ich meine auch, ... dass der Computers in der Grundschule noch nicht zu intensiv eingesetzt werden sollte, da es Bereiche zu fördern gibt, für die der PC ungeeignet ist (z.B. Feinmotorik). Auch muss ja gezielt (Schreibung von Wörtern, Grammatik usw.) geübt werden, was ohne PC erfolgen kann, Schreiben mit der Hand, da das Geschriebene eine Vertiefung erfährt usw., usw.). (...) Den PC aus Grundschulklassen zu verbannen ist meines Erachtens aber nicht nötig. Ich glaube sogar, dass der Einsatz manchmal sinnvoll ist. Es geht doch um die Integration von PCs, nicht darum "konventionellen" Unterricht in der Breite zu ersetzen.“ (Luise)

„Die, die professionell schreiben, haben es aber auch erst handschriftlich gelernt. "Zu früh" ist demnach vielleicht hier das zu gebende Stichwort. Die Schüler müssen ja auch erst etwas Sicherheit in Rechtschreibung usw. erlangen. die Rechtschreibprüfung kann das ja nicht ersetzen. Erst sollte die Handschrift angefangen werden zu trainieren und dann kann sich dem Tippen am PC zugewendet werden. (...) das Handschriftliche sollte nicht vernachlässigt werden. Denn das Schreiben mit der Hand ist unter anderem Ausdruck der Persönlichkeit eines Menschen. Außerdem beschleunigt das Schreiben per Hand den Lernprozess, denn durch das Schreiben erfährt das Gelesene und Gedachte eine Vertiefung.“ (Edith)

„(...) ob der Einsatz von Rechnern in der Grundschule angemessen ist. Ich denke dies wäre nur wenig angemessen. (...) In der Grundschule sollten zunächst die Grundfertigkeiten (Schreiben, Rechnen, Lesen ...) gefördert werden und zwar auf konventionelle Weise. Es soll nicht nur Wissen als solches angeeignet werden – die Schüler sollen sich weiterhin auch Verhaltensweisen aneignen, die auch in Zukunft ein angemessenes bzw. effektives Arbeitsverhalten ermöglichen (Hierzu zählt besonders der angemessene Umgang mit traditionellen Arbeitsmaterialien). Zudem halte ich die Schulung der Feinmotorik der Schüler gerade in den schulischen Anfangsjahren für sehr wichtig. Diese ist meiner Meinung nach eher durch traditionelle Schreibtätigkeiten etc. zu schulen als durch die Arbeit mit dem Computer. (...) Es sollte nicht jeder angehende Lehrer dazu "gezwungen" werden sich mit technischen Medien im Detail auseinanderzusetzen. Natürlich sollte jeder Basiswissen besitzen, aber mehr auch nicht – wenn er nicht will! Dieses Basiswissen muss nicht in der Uni erworben werden! Zudem, so denke ich, muss man nicht überdurchschnittlich medienkompetent sein, um ein guter Lehrer sein zu können! Wir sind in erster Linie Wissensvermittler (fachbezogen) mit Erziehungsauftrag und keine Informatiker.“ (Dora)

„Da in der Grundschule erste Erfahrungen mit Schrift gemacht werden, finde ich, dass zunächst die HAND-Schrift geübt werden sollte. Im ersten Schuljahr sollte meiner Meinung nach ein PC nur zum Übertragen von eigenen, selbst geschriebenen Texten genutzt werden. (...) In den Folgejahren ist das Schreiben am PC oft eine willkommene Abwechslung zum oft noch anstrengenden "mit der Hand schreiben" – sollte allerdings nicht Überhand gewinnen, bevor die Handschrift nicht gefestigt ist. Mit dem Smartboard könnte man dieses "Problem" bedingt umgehen, da direkt am Board die Handschrift geübt werden kann.“ (Barbara)

„Ich habe eigentlich keine Bedenken gegenüber Schreibprozessen am Computer ... solange man das Schreiben nicht auf den Computer reduziert! Ich denke, das Schreiben mit der Hand darf auf keinen Fall in den Hintergrund gedrängt werden (...) Ich finde die Einwände gegen die Technisierung der Schule ja auch durchaus berechtigt. Aber ich verstehe die ganzen technischen Möglichkeiten auch absolut als Ergänzung! Keiner sagt, dass nur noch mit technischen Medien unterrichtet werden soll und keiner sagt, dass man auf z. B. "Schreiben mit der Hand" verzichten soll! ich finde ehrlich gesagt, dass heutzutage beides absolut seine Berechtigung hat und beides in der Schule berücksichtigt werden muss.“ (Gerda)

Insgesamt ließen sich diese Einwände der Studierenden so zusammenfassen:

A

- Der Erwerb der Schriftlichkeit gelingt nicht am PC.
- Kinder müssen erst einmal handschriftlich schreiben lernen. Die Handschrift ist der Vorläufer. Das Erlernen der Handschrift ist die Voraussetzung fürs Schreiben. Das Schreiben am PC darf nicht zu früh einsetzen. Das Schreiben am PC darf nicht geschehen, bevor die Handschrift gefestigt ist.
- Schreiben darf nicht auf das Schreiben am PC reduziert werden, das Schreiben mit der Hand darf auf keinen Fall in den Hintergrund gedrängt werden.

B

- Gescheite Schriftlichkeit gelingt nur, wenn per Hand geschrieben wird. Texte müssen immer erst handschriftlich verfasst werden und können dann auf den PC übertragen werden.
- Schreiben mit der Hand fördert den Lernprozess, denn das Gelesene und Gedachte wird vertieft.

C

- Das Schreiben mit der Hand ist Ausdruck der Persönlichkeit eines Menschen.

D

- Die Handschrift darf nicht vernachlässigt werden, man wird sonst so faul. Schüler sollen sich die Mühe machen, mit der Hand zu schreiben.
- Lehrer darf man nicht dazu zwingen, die Schüler am PC schreiben zu lassen.

Die Studierenden begründen ihre Abwehr in der Regel nicht mit Forschungsergebnissen. Ihre Einwände haben eher den Rang von Alltagstheorien. Ihnen gälte es, offensiver entgegenzutreten, z. B. mit Argumenten, die die fachdidaktische Forschung schon bereithält. Und es gälte, diese Argumente verstärkt zu publizieren.

4 Bildungssoftware statt Lernsoftware

Solange das Medialitätsbewusstsein der Studenten nicht geweckt wird, solange sie nicht verstehen und akzeptieren, dass Schreiben schon immer eine technologische Basis hatte, solange Schreiben als ein natürlicher Prozess verstanden wird, solange bleiben digitale Medien nur schmückendes Beiwerk im Deutschunterricht, Sahnehäubchen und nichts weiter.

In einer Zeit, in der sich sowohl der Beschreibstoff, der *Griffel* und die *Tinte* entmaterialisiert haben – weil Schreiben heute Schreiben am Computer bedeutet, jedenfalls soweit es sich um Schreibvorgänge im öffentlichen Leben, in Produktion, Verwaltung und Dienstleistung handelt, in einer Zeit, in der die meisten Schriftstücke, Formulare etc. digitalisiert sind, Buchmanuskripte ebenso wie Zeitungsartikel oder Gesetzestexte – hat Schreiben mit der Hand seinen Ort nur noch im privaten oder künstlerischen Raum. Die Schule hat sich noch nicht darauf eingestellt, obwohl es in der Deutschdidaktik eine Reihe von Publikationen gibt, die sich in den letzten Jahren verstärkt mit der Veränderung des Schreibprozesses durch die digitalen Medien auseinandergesetzt haben.

Ein wesentlicher Vorzug von Schreibprozessen im digitalen Medium ist die Verbesserung der Bedingungen für Überarbeitungs- und Korrekturprozesse (Schreibkonferenzen u. Ä.). Hier soll nur auf einen wichtigen Aspekt der Korrektur verwiesen werden: den orthographischen. Liegt ein Text digitalisiert vor, dann kann der Schreiber bei der orthographischen Korrektur von auf Korrektur spezialisierter Software – z. T. in Textverarbeitungssoftware enthalten, z. T. separat einbindbar – unterstützt werden: ABC-Prüfung von Word, OpenOffice oder DudenKorrektor 3.51 u.a. Darüber hinaus lassen sich digitale Rechtschreibwörterbücher mit einbinden, natürlich auch Fremd-, Etymologie- und Synonymwörterbücher (z. B. die OfficeBibliothek des Dudenverlags.) [Be07b]

In Bildungsprozessen aber erschöpft sich die Auseinandersetzung mit dem jeweiligen Instrument nicht mit seiner Handhabbarkeit. Der zweckvolle Einsatz des Instruments ist nur ein Beispiel für einen möglichen Umgang mit demselben; das Ziel ist Erkennen und Verstehen.

Die Aneignung des Gegenstands und seiner Funktionalitäten ist der Weg dazu. Wenn *Werkzeuge* in Bildungsprozessen eingesetzt werden sollen, dann nicht als *Werkzeuge* zur Herstellung eines Produkts oder zur Durchführung einer Operation, auch nicht als Instrument, wie z. B. ein Musikinstrument, sondern als ein Medium, mit dem der Mensch versucht, sich Wissen und Verständnis über etwas anzueignen. Der Lernerfolg ist dann nicht an der Beherrschung des Instruments zu messen.

Als Ergebnis eines Dialogs zwischen Informatikern und Didaktikern wäre also auf Software zu hoffen, die die automatisierten Funktionen nicht verbirgt, sondern sie sichtbar und zugänglich macht für den Lernprozess. Werden Software-Tools in Bildungsprozessen eingesetzt, ist es entscheidend, dass Schüler die Ergebnisse nicht nur auf ihre Tauglichkeit hin überprüfen können, sondern auch ihr Zustandekommen erkennen, dass sie verfolgen können, welche Prozeduren die Software abarbeitet im Vergleich zu den Schritten, mit denen sie als Schüler Probleme lösen. Eine solche Software wäre dann zugleich ein Bildungsmedium. [vgl. Sch07]

Literaturverzeichnis

- [Au04] Auditierung: Förderprogramm Neue Medien in der Bildung, Förderbereich Schule. Abschlussbericht des Arbeitskreises unter Vorsitz von Bardo Herzig, DLR Projektträger Neue Medien in der Bildung + Fachinformation Sankt Augustin, November 2004 http://www.dlr.de/pt_nmb/Projektauditierung_Schule (10.10.2007)
- [Be07a] Berndt, E.-B.: Warum die Werkzeug-Metapher in Bildungsprozessen in die Irre führt. In: OBST Heft 72, 2007; S. 109-133
- [Be07b] Berndt, E.-B.: Warum Fachdidaktiker den Dialog mit der Informatik suchen sollten – am Beispiel der Rechtschreibkorrekturprogramme. In: (Jonas, H.; Josting, P. Hrsg.): Jahrbuch Medien im Deutschunterricht 2007, KopaedVerlag, München, 2007; S. 136–152,
- [BI93] Berndt, E.-B.; Ihmels, G: Die Elektronische Wandtafel – ein didaktisches Konzept zum computerunterstützten Unterricht. In (Troitzsch, K.G. Hrsg.): Informatik als Schlüssel zur Qualifikation. GI Fachtagung “Informatik und Schule Koblenz 1993. Springer, Berlin Heidelberg, 1993; S. 156-161.
- [Bu06] Bundesministerium für Bildung und Forschung (Hrsg.) IT-Ausstattung der allgemein bildenden und berufsbildenden Schulen in Deutschland. Bonn, Berlin 2006

- http://www.bmbf.de/pub/it-ausstattung_der_schulen_2006.pdf 7.3.2009
- [Bu09] Bundesministeriums für Bildung und Forschung (BMBF): Digitale Schule – wie Lehrer Angebote im Internet nutzen. Eine Bestandsaufnahme im Auftrag des MMB-Institut für Medien- und Kompetenzforschung, Essen 2008
www.dlr.de/pt/ResourceImage.aspx?rid=43167 7.3.2009
- [DG07] Deutscher Germanistentag. Natur – Kultur Universalität und Vielfalt in Sprache, Literatur und Bildung. Philipps-Universität Marburg 23. bis 26. September 2007. Sektion 4: Medialität von Sprache, Literatur und Kunst.
<http://www.germanistenverband-hochschule.de/kev/germtag07/index.php?id=127>.
3.2009
- [GAL07] Gesellschaft für Angewandte Linguistik: Jahrestagung vom 26.9.- 28.9.2007 Universität Hildesheim. Themenbereich „Sprachbasierte Medienkompetenz von Kindern und Jugendlichen“ (26.9./27.9.2007).
<http://www.gal-ev.de/jahrestagung-2007.html> 07.03.2009
- [Ha09] Hanselmann, U. »Ein bisschen Word« In: DIE ZEIT, ZEIT ONLINE 2009 19.02.2009 Nr. 09 <http://www.zeit.de/2009/09/C-Informatik> (7.3.2009)
- [HG06] Herzig, B., Gräfe, S.: Digitale Medien in der Schule. Standortbestimmung und Handlungsempfehlungen für die Zukunft. Bonn: Deutsche Telekom und Universität Paderborn 2006;
www.schulen-ans-netz.de/san/10jahreschulenansnetz/dokus/ATT43351.pdf 7.3.2009
- [Ifib07] Ifib – Institut für Informationsmanagement: Nutzung digitaler Medien in den Schulen im Bundesland Bremen. Ergebnisse und Vergleich der Befragung von Schulen, Lehrkräften sowie Schülerinnen und Schülern aus dem Frühjahr 2006, Bremen, 2007;
http://www.ifib.de/index.php?npoint=1,0,0&sprache=de&ansicht=klein&monats_id=45
7.3.2009
- [Ju 07] Jung, E.: Der Laptop-Flop. In: Süddeutsche Zeitung vom 10.05.2007
<http://www.sueddeutsche.de/computer/990/320860/text/> 7.3.2009
- [SI02] Schaumburg, H., Issing, L.J.: Besser Lernen durch Laptops? Ergebnisse eines Modellversuchs zur Nutzung mobiler Computer in der Schule.: Verlag Bertelsmann Stiftung, Gütersloh 2002
http://www.bertelsmann-stiftung.de/cps/rde/xchg/SID-0A000F14-11E4408F/bst/hs.xsl/downloads.htm?suchrubrik=bildung&suchbegriff=Modellversuch+zur+Nutzung+mobiler+Computer+in+der+Schule&Suche_absenden.x=23&Suche_absenden.y=12&innerRequest=true# Zugriff: 10. 10.2007
- [Sch07] Schelhowe, H.: Technologie, Imagination und Lernen Grundlagen für Bildungsprozesse mit Digitalen Medien. Monografie. Waxmann-Verlag Münster 2007
- [Tr93] Troitzsch, K.G. Hrsg.: Informatik als Schlüssel zur Qualifikation. GI Fachtagung “Informatik und Schule 1993, Koblenz 1993. Springer, Berlin Heidelberg, 1993
- [VdS07] VdS Bildungsmedien e.V.: PISA-Reformen ohne neue Bildungsmedien Umsatz mit Schulbüchern stagniert; Umsatz mit Unterrichtssoftware sogar rückläufig. Frankfurt/Köln, 28.02.2007
<http://bildungsklick.de/pm/51563/pisa-reformen-ohne-neue-bildungsmedien>
- [Wa04] Wagner, W.-R.. Medienkompetenz revisited: Medien als Werkzeuge der Weltaneignung: ein pädagogisches Programm. München: Kopaed Verlag 2004
- [Wa08] Wagner, Wolf-Rüdiger: Medialitätsbewusstsein. In: Computer und Unterricht, 70, 2008; S. 34-36

Zahlen, Daten und Fakten zum Informatikunterricht an den Gymnasien Österreichs

Peter Micheuz

Institut für Informatik Systeme
Universität Klagenfurt
Universitätsstraße 65-67
9020 Klagenfurt
peter.micheuz@uni-klu.ac.at

Abstract: Informatikunterricht wird an den allgemein bildenden höheren Schulen Österreichs in unterschiedlicher Quantität und Qualität in allen Schulstufen erteilt. Die Heterogenität als Folge von Schulautonomie ist gerade in diesem Bereich beispiellos. Basierend auf statistischen Elementardaten sowie den Ergebnissen einer nationalen Online-Umfrage an allen Gymnasien, die der Autor im Jahr 2007 durchgeführt hat, werden exemplarisch Einblicke in die gegenwärtige Lage zur Schulformatik in diesem Schultyp gewährt.

1 Vorbemerkung

In allen 330 Gymnasien Österreichs findet im Schuljahr 2008/2009 lt. Auskunft des BMUKK (Bundesministerium für Unterricht, Kunst und Kultur) Informatikunterricht im Ausmaß von ca. 11.400 Wochenstunden statt. Davon entfallen etwa ein Drittel auf die Sekundarstufe I (5.–8. Schulstufe mit insgesamt ca. 115.000 SchülerInnen) und zwei Drittel auf die Sekundarstufe II (9.–12. Schulstufe mit insgesamt 90.000 SchülerInnen). Von den insgesamt 1,2 Millionen SchülerInnen Österreichs besucht ein Sechstel den Schultyp AHS (Allgemeinbildende Höhere Schule, Gymnasium), der laut Schulorganisationsgesetz die Aufgabe hat, den SchülerInnen eine umfassende und vertiefte Allgemeinbildung zu vermitteln und sie zugleich zur Hochschulreife zu führen. Aus dem Unterrichtsbudget werden vom österreichischen Staat ungefähr 20 Millionen Euro für ca. 2000 Informatiklehrende aufgewendet.

Wenn Budgets die in Zahlen gegossene Politik eines Staates sind, dann gilt das in besonderer Weise für die Bildungspolitik. Innerhalb aller gymnasialen Fächer beträgt der Anteil an Informatikstunden rund drei Prozent. Zwar wissen wir nun, was der Informatikunterricht kostet, aber was ist er wert? Diese erkenntnistheoretische Frage zu beantworten ist objektiv unmöglich. In diesem Beitrag aber soll wenigstens versucht werden, auf Basis der erwähnten Studie über die reine Kostenfrage hinaus exemplarisch Antworten darauf zu geben, wie Informatikunterricht an Österreichs Gymnasien organisiert und gesteuert wird, welche Inhalte vermittelt werden und wie er wahrgenommen wird.

2 Rahmenbedingungen

An den AHS findet der Informatikunterricht in mehreren Ausprägungen statt. Zurückzuführen ist diese Vielfalt nicht zuletzt auf die in den 90er Jahren begonnene und nach den letzten Lehrplanreformen 1999 und 2003 verstärkte Dezentralisierung und Autonomisierung. Im verpflichtenden Kernbereich der Sekundarstufe I ist kein verbindlicher Informatikunterricht und damit auch kein zentraler Lehrplan vorgesehen. Dennoch wurden schulautonom ca. 2200 Informatik-Pflichtstunden mit schulautonomen Lehrplänen auf Kosten anderer Fächer implementiert. Dazu kommen noch Freigegegenstände und Unverbindliche Übungen im Ausmaß von ca. 1500 Wochenstunden. Dieses Angebot streut allerdings enorm. Das Spektrum reicht von Schulen mit einem Nullangebot bis zu zwei Stunden pro Schulstufe in Gymnasien mit informatischem Schwerpunkt. Der fiktive, durchschnittliche AHS-Schüler erhält im Laufe der Sekundarstufe I zwischen ein und zwei Stunden formellen Informatikunterricht. In diesem Beitrag erfahren wir allerdings, dass mindestens ein Drittel der AHS-SchülerInnen die Sekundarstufe I ohne formalen Informatikunterricht absolviert.

Seit dem Jahr 1985 stellt das Unterrichtsfach Informatik [Re03] in der 9. Schulstufe am Beginn der Sekundarstufe II die einzige Invariante informatischer Unterrichtserteilung an Österreichs Gymnasien dar. Es ist für alle SchülerInnen im Ausmaß von zwei Wochenstunden verpflichtend. Anschließend kann Informatik in der 10. bis zur 12. Schulstufe als Wahlpflichtfach im Ausmaß von vier bis sechs Wochenstunden gewählt und auch mündlich maturiert werden. Derzeit wählt ungefähr ein Fünftel der SchülerInnen der 10.-12. Schulstufe dieses Wahlpflichtfach. Ein Anteil, der im Vergleich zu den Vorjahren eine signifikante Abnahme darstellt. Ebenso rückläufig ist die Anzahl der in Informatik abgelegten Reifeprüfungen.

Nicht nur in der Sekundarstufe I herrscht bezüglich des Informatikangebots eine große Vielfalt, die sich bereits seit langem dem ministeriellen Überblick entzogen hat, sondern auch in der Sekundarstufe II. Hier hat die Schulautonomie seit der „AHS Oberstufe neu“ ab 2003 zu einer zusätzlichen Unübersichtlichkeit geführt. Diese ist vor allem auf die facettenreichen IT/Informatik Schwerpunktsetzungen in ca. einem Drittel aller Gymnasien in der Sekundarstufe II zurückzuführen.

3 Lehrpläne und ineffiziente Inputsteuerung

Da für die Sekundarstufe I im verbindlichen Lehrfächerkanon kein Informatikunterricht vorgesehen ist, gibt es für die 5.-8. Schulstufe auch keine Informatik-Lehrpläne und somit kein elaboriertes Framework für Informatik-Inhalte in diesen Altersstufen. Ein kurzer inhaltlicher Hinweis für die Unverbindliche Übung „Einführung in die Informatik“ zur allgemeinen Interessen- und Begabungsförderung findet sich in [LP99]: “Die Schülerinnen und Schüler sollen Sicherheit in der Bedienung von Computern samt Peripheriegeräten, Geläufigkeit bei der Verwendung üblicher Anwendersoftware und grundlegende Kompetenzen im Umgang mit neuen Technologien insgesamt gewinnen und interessenorientierte Arbeiten mit neuen Technologien sowohl individuell als auch im Team durchführen können.“ Diese gesetzliche Grundlage für 1450 Wochenstunden gilt

in der Sekundarstufe I für 260 Gymnasien. Im Schnitt bedeutet dies zwei bis drei Informatik-Kurse pro Schule für alle vier Jahrgänge, die sowohl quantitativ als auch schulstufenbezogen unter den Schulen heterogen verteilt sind.

Auf die ineffektiven Versuche, informationstechnologische Grundbildung seit Beginn der 90er Jahre durch Integration in den „Trägerfächern“ Mathematik, Deutsch, Englisch und Geometrisches Zeichnen zu implementieren, soll hier nur verwiesen werden [Ha97]. In der Lehrplannovelle 1999 für die Sekundarstufe I findet sich bei den allgemeinen didaktischen Grundsätzen der Passus „Den neuen Technologien kommt verstärkt Bedeutung zu. Dies gilt in besonderem Maße für die Oberstufe der allgemein bildenden höheren Schule. Hier sind in allen Gegenständen Informationsmanagement sowie Lern- und Unterrichtsorganisation mit Mitteln der Informationstechnologie zu praktizieren.“

Kann in der in der Sekundarstufe I noch von einer Empfehlung für den Einsatz Digitaler Technologien gesprochen werden, so muss in der Sekundarstufe II (Oberstufe) von einem unmissverständlichen Verbindlichkeitscharakter gesprochen werden, der auch in den Fachlehrplänen klar zum Ausdruck kommt. Nicht zufällig findet sich unter den didaktischen Grundsätzen zum Lehrplan des Pflichtfaches Informatik in der 9. Schulstufe (2 Wochenstunden) folgender Passus: „Der Informatikunterricht muss beispielhaft für den sinnvollen Einsatz verfügbarer Technologien sein.“ In diesem Aufsatz wird auch der Frage nachgegangen, ob der gegenwärtige Informatikunterricht dieser Vorbildfunktion nachkommt bzw. nachkommen kann. Mehr als zwei Drittel, das entspricht 4900 von insgesamt 7200 Wochenstunden Informatik in der gesamten Oberstufe, fallen auf dieses Pflichtfach Informatik, das als österreichisches Spezifikum im laufenden Schuljahr 25 Jahre alt geworden ist.

Mehr als die Wurzeln und die Entwicklung dieses Faches (vgl. [Mi03]), dessen Lehrplan aus dem Jahr 1985 erst 2003 im Zuge der „AHS Oberstufe neu“ geändert wurde, interessiert hier die inhaltliche Umsetzung. Dabei interessiert die Antwort auf die Frage, ob der Informatikunterricht in der 5. Klasse lediglich als Supportfunktion für andere Fächer zu sehen ist, oder ob den SchülerInnen ein umfassendes Bild von Informatik vermittelt wird. Dies gilt vor allem für jene SchülerInnen, für die diese Schulstufe das einzige Zeitfenster formellen Informatikunterrichts am Gymnasium darstellt. In den „Didaktischen Grundsätzen“ wird besonders auf die Übergangsphase von der 8. zur 9. Schulstufe aufmerksam gemacht: „Im Informatikunterricht sind besondere didaktische Überlegungen anzustellen, um Defizite auszugleichen und individuelle Stärken einzubinden.“

Der Lehrstoff für die 9. Jahrgangsstufe ist kompakt genug, um hier in ungekürzter Form wiedergegeben zu werden. Die Schülerinnen und Schüler sollen:

- *Informationsmanagement und Lernorganisation für die eigene Lernarbeit und Weiterbildung mit geeigneter Software in der Praxis umsetzen und dabei vorhandene Informationsquellen erschließen und unterschiedliche Informationsdarstellungen ausgehend von den Vorkenntnissen anwenden*
- *Inhalte systematisieren und strukturieren sowie Arbeitsergebnisse zusammenstellen und multimedial präsentieren können*

- *ein vernetztes Informationssystem für die individuelle Arbeit aufbauen und nutzen können*
- *den sicheren Umgang mit Standardsoftware zur schriftlichen Korrespondenz, zur Dokumentation, zur Publikation von Arbeiten, zur multimedialen Präsentation sowie zur Kommunikation erreichen*
- *Kalkulationsmodelle erstellen und die Ergebnisse bewerten und interpretieren können; eine einfache Datenbank benutzen können*
- *Einblicke in wesentliche Begriffe und Methoden der Informatik, ihre typischen Denk- und Arbeitsweisen, ihre historische Entwicklung sowie ihre technischen und theoretischen Grundlagen gewinnen und Grundprinzipien von Automaten, Algorithmen und Programmen kennen lernen*
- *wesentliche Maßnahmen und rechtliche Grundlagen im Zusammenhang mit Datensicherheit, Datenschutz und Urheberrecht kennen lernen sowie die Auswirkungen des Technikeinsatzes auf die Einzelnen und die Gesellschaft nachvollziehen*
- *Einsatzmöglichkeiten der Informatik in verschiedenen Berufsfeldern kennen lernen und somit in ihrer Berufsorientierung Unterstützung finden*

Zwei Lehrplankomentierungen zu diesen eher abstrakten Zielvorgaben bieten [Mi03] und [Ca05]. Auf eine weitere tiefeschürfende Inhaltsanalyse wird hier zugunsten exemplarischer Ergebnisse einer nationalen empirischen Studie verzichtet, welche vor allem die Erforschung der Praxis des Informatikunterrichts in der 9. Jahrgangsstufe (5. Klasse AHS) zum Ziel hatte.

Um das durch Lehrpläne öffentlich zugängliche, intendierte Bild der Informatik für die Sekundarstufe II der AHS zu komplettieren, wird hier der Lehrstoff für das Wahlpflichtfach Informatik in der 10.-12. Schulstufe angegeben. Dieser besteht lediglich in einer wahllosen, fast beliebigen Aufzählung folgender informatischer Themenbereiche:

Grundprinzipien der Informationsverarbeitung, Konzepte von Betriebssystemen, Aufbau und Funktionsweise von Netzwerken, Datenbanken, Lern- und Arbeitsorganisation, Konzepte von Programmiersprachen, künstliche Intelligenz, Erweiterung der theoretischen und technischen Grundlagen der Informatik, grundlegende Algorithmen und Datenstrukturen, Informatik, Gesellschaft und Arbeitswelt, Rechtsfragen.

Es ist nicht Ziel dieser Arbeit, diese Form der Inputsteuerung zu kommentieren. Die Mehrheit der befragten und involvierten InformatiklehrerInnen, so eines der überraschenden Erkenntnisse aus der Studie, ist mit diesen vagen Vorgaben einverstanden.

Ein spezielles Kapitel, auf das in diesem Beitrag nicht näher eingegangen wird, ist die Reifeprüfung aus Informatik. Diese erfolgt nur mehr in ganz wenigen Schulen schriftlich in Form einer Klausurarbeit. In welchem Ausmaß SchülerInnen Informatik als mündliches Reifeprüfungsgebiet wählen und wie viele sich für eine vorwissenschaftliche Fachbereichsarbeit entscheiden, muss noch in einer weiteren Studie empirisch erhoben werden.

4 Zur empirischen Feldforschung und zum Design der Studie

Das Fehlen eines effizienten Bildungsmonitorings als Grundlage für evidenzbasierte Bildungssteuerung darf als eine Schwachstelle österreichischer Schulpolitik bezeichnet werden. Dies trifft auch auf den Informatikunterricht zu, wo eine solide Datenlage nicht vorhanden ist. Eine österreichweite Feldstudie, die vom Autor in der Zeit von März bis Mai 2007 in Form eines Online Surveys durchgeführt wurde, konnte etwas Abhilfe schaffen. Diese Studie wurde unter dem Titel „Über Trends und den Stand des Informatik-Unterrichts an den AHS in Österreich - eine Feldstudie“ über Projektmittel der IMST-Initiative des Unterrichtsministeriums („Innovationen machen Schulen top“) gefördert.

Werden Studien an Schulen durchgeführt, so spielen so genannte „Gate Keepers“ [vgl. Gr99], vor allem die SchulleiterInnen und die nationale Behörde, eine wichtige Rolle. Vom Unterrichtsministerium wurde ein Erlass über die Genehmigung zur Durchführung der Befragung eingeholt. Die Schlüsselrolle in der Durchführung dieser Online-Studie spielten die Schulleitungen. Von ihnen hing es ab, im Sinne eines „informed consent“ [vgl. BD95] die Umfrage an den einzelnen Schulen zu unterstützen und möglichst viele InformatiklehrerInnen zum Mitmachen zu bewegen.

Der Zeitpunkt für eine Online-Befragung in dieser Größenordnung schien günstig, da einerseits alle Gymnasien Anfang 2007 bereits gut an das Internet angebunden waren und mit einer großen „Befragungsrenitenz“ noch nicht zu rechnen war und es in dieser Zeit noch wenige Online-Untersuchungen gab. In der Vorbereitungsphase wurde für diesen Zweck eine Website (<http://ahs.schulinformatik.at>) mit vier zielgruppenorientierten, webbasierten Fragebögen für vier Populationen eingerichtet, nämlich für alle AHS-Schulleitungen Österreichs, alle AHS-InformatiklehrerInnen, alle SchülerInnen aus den 5. Klassen (9. Schulstufe) sowie für eine Stichprobe von SchülerInnen der 10. Schulstufe. Nach umfangreichen Vorbereitungsarbeiten einschließlich der Konzeption der Fragebögen, der technischen Umsetzung der Online-Befragung auf einem Webserver (Internetadresse siehe oben) und dem Verfassen der Einladungsscheiben an alle Zielgruppen, wurden Anfang März 2007 die Schulleitungen aller 330 Gymnasien Österreichs postalisch eingeladen, sowohl selbst an der Befragung teilzunehmen als auch alle InformatiklehrerInnen ihrer Schule zum Mitmachen zu bewegen. Von diesen hing es wiederum ab, allen SchülerInnen der 9. Schulstufe die Beantwortung des Schülerfragebogens zu ermöglichen. Schließlich wurden die Klassenvorstände aller Klassen der 10. Schulstufe gebeten, einer Stichprobe von zufällig gewählten SchülerInnen der 10. Schulstufe eine Teilnahme an der Befragung zu ermöglichen.

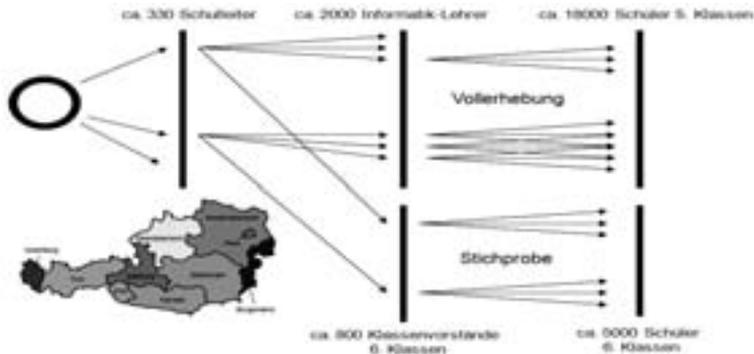


Abbildung 1: Die Zielgruppen der nationalen Online-Umfrage

Die Datenerhebung erfolgte online nach dem Versand von Einladungsbriefen an die Schulleitungen mit insgesamt fast 30.000 Zugangskennungen für alle 4 Kohorten. Die ersten Antworten langten Mitte März 2007 ein, und Mitte Mai wurde die Umfrage abgeschlossen. Die Rücklaufquoten in der Größenordnung von fast 25% und insgesamt 9280 Datensätzen aus vier zielgruppenorientierten Online-Fragebögen mit mehr als 200 abgefragten Items dürfen als zufriedenstellend bezeichnet werden. Diese konnten nur durch ein nochmaliges Nachfassen in Form eines Erinnerungsschreibens nach den Osterferien 2007 erreicht werden. Es gab Rückmeldungen von 169 Schulen, das entspricht mehr als der Hälfte aller Gymnasien. Verständlicherweise kann es nicht Ziel sein, hier detailliert auf die umfangreichen Fragebögen einzugehen. Weitere Informationen zur Umfrage sowie deskriptive Auswertungen können auf der Website <http://ahs.schulinformatik.at> eingesehen werden.

5 Exemplarische Ergebnisse im Überblick

5.1 Rahmenbedingungen und Daten zur IT-Infrastruktur

Wenig überraschend findet Informatikunterricht (auch) an den AHS Österreichs bis auf zu vernachlässigende Ausnahmefälle ausschließlich an Computern in Informatikräumen statt. Alle Gymnasien verfügen über mindestens zwei Computersäle, die Informatikunterricht praxisorientiert in Gruppengrößen von 5 bis 15 SchülerInnen ermöglichen. Eine in dieser Studie inkludierte Infrastrukturerhebung offenbart allerdings große Unterschiede sowohl in Quantität als auch in der Qualität der IT-Ausstattung. So variieren die Quotienten Computer : SchülerInnen an den Gymnasien von 1:5 bis zu 1:20. Schwerpunktschulen verfügen über bis zu 5 Informatiksäle. Laptopklassen (in weniger als 5% der AHS) stagnieren auf niedrigem Niveau. Gab es ab 1985 für alle AHS noch zentrale bundesweite Ausstattungsinitiativen mit Sonderbudgets, so müssen seit Jahren IT-Investitionen aus dem Schulbudget finanziert werden. Mehr als zwei Drittel aller AHS-Schulleiter beklagen, dass die IT-Infrastruktur unter den derzeitigen Bedingungen nur schwer aufrechtzuerhalten ist. Fast ein Drittel der Schulen greift zur Selbsthilfe und ergänzt das IT-Budget durch Sponsoring und (moderate) jährliche SchülerInnenbeiträge.

Diese zusätzlichen Investitionen sind vor allem in Schulen unabdingbar, die im Normalunterricht Digitale Medien verstärkt einsetzen. Als zufrieden stellend darf die Internetanbindung bezeichnet werden, die in Bandbreiten ab 2 Mbit an den AHS zum Standard gehört.

5.2 Informatik an den AHS in der Sekundarstufe I

Der Heterogenität der IT-Infrastruktur steht das differenzierte Angebot des Informatikunterrichts in der Sekundarstufe I um nichts nach. Je nach Profilbildung und schulautonomer Schwerpunktsetzung variiert das Angebot extrem, wobei sich kaum Schulen mit der gleichen Angebotsstruktur finden.

Differenziertes Angebot in der Sekundarstufe (79 Schulen)	
8. Schulstufe	P U P U U U P U Z Z U P U U Z U P P P F F P U P U U U P P P U U F P U U Z U F U P P P U U P Z F
7. Schulstufe	P P U P P P U U P P U P U U U U U P F P U P U U U U P U U U F P U U U U Z U F U P P P U P U P U F
6. Schulstufe	P P P U U U P P P P U F U P U U U U U P P P P U P Z Z U F P P P P P
5. Schulstufe	U F P U P U P U U P U Z Z P P U P F P Z P Z U U Z P P P P P Z Z P Z P P P P F

Abbildung 2: Informatikangebot in der Sekundarstufe I der AHS

Abb. 2 veranschaulicht den organisatorischen „Fleckerlteppich“ der rückmeldenden Gymnasien. In der Tabelle symbolisiert P das schulautonome Pflichtfach Informatik, F das Freifach, U die Unverbindliche Übung und Z einen zeitlich begrenzten IT-Kurs. Insgesamt 820 Wochenstunden für ca. 80 Schulen und 1200 Klassen in der Sekundarstufe ergeben ein durchschnittliches Angebot von etwa 10 Wochenstunden Informatik in der Sekundarstufe I in einer fiktiven Durchschnittsschule. Wenn man die Klassenteilungen mit maximal 15 SchülerInnen pro Gruppe berücksichtigt, findet pro Jahr in einem Drittel der Klassen formeller Informatikunterricht statt.

Aus Sicht der SchülerInnen der 5. Klassen (9. Jahrgangsstufe) sieht das wahrgenommene Angebot in der Sekundarstufe I (rückblickend) sehr differenziert aus. Mehr als ein Drittel aller SchülerInnen nahm überhaupt kein formelles Informatikangebot wahr, jeweils ein Fünftel gab an, zwei bzw. vier Stunden Informatikunterricht gehabt zu haben. 140 SchülerInnen von insgesamt 5100 rückmeldenden Probanden nahmen an acht Wochenstunden Informatikunterricht teil.

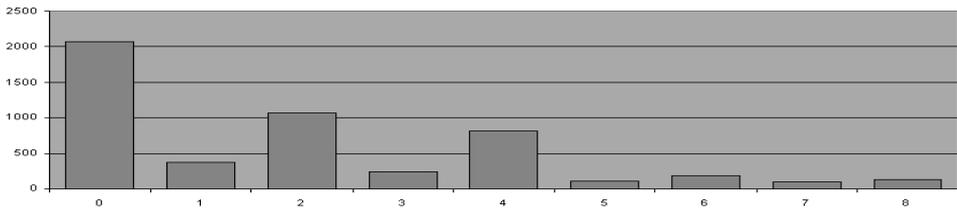


Abbildung 3: Informatikstunden in der Sekundarstufe I aus SchülerInnenperspektive

Auf eine Detailanalyse bezüglich geschlechtsspezifischer und regionaler Unterschiede wird hier verzichtet, ebenso wie auf den Versuch, im Rahmen der nationalen Studie eine Inhaltsanalyse hunderter schulautonomer Lehrpläne durchzuführen. Regionale Studien und das Erfahrungs- und Überblickswissen des Autors lassen einen langjährigen Trend zur reinen Bediener- und Werkzeugschulung in Anlehnung an die Zielsetzungen des Europäischen Computerführerscheines (ECDL) erkennen.

Die Frage, ob ein „eigenständiges Fach IT/Informatik in der Sekundarstufe I für alle SchülerInnen notwendig sei“, bejahten fast zwei Drittel aller SchülerInnen. Nicht ganz die Hälfte gab an, sich viele Computerkenntnisse selbst angeeignet zu haben.

Der Lernzuwachs in der Sekundarstufe I lässt sich über die Frage „Ich habe im Informatikunterricht viel/wenig/nichts gelernt“ grob abschätzen (siehe Abb. 4). Abgesehen von der überwiegenden Mehrheit ohne formellen Informatikunterricht fällt auf, dass in Gymnasien mehr als die Hälfte der SchülerInnen angab, im Informatikunterricht wenig gelernt zu haben. Für ca. 900 ehemalige HauptschülerInnen bei der Befragung, die sich für ein Oberstufengymnasium entschieden haben, fiel die Antwort „viel/wenig gelernt“ eher balanciert aus. Es fällt auch auf, dass nur ca. 10% der Hauptschüler keinen Informatikunterricht hatte. Eine weitere Analyse der vorliegenden Daten lässt den eindeutigen Schluss zu, dass das Informatik-Angebot an Hauptschulen fast doppelt so groß ist wie das in der Sekundarstufe der Gymnasien. In Abb. 4 repräsentieren die dunklen Balken die HauptschülerInnen.

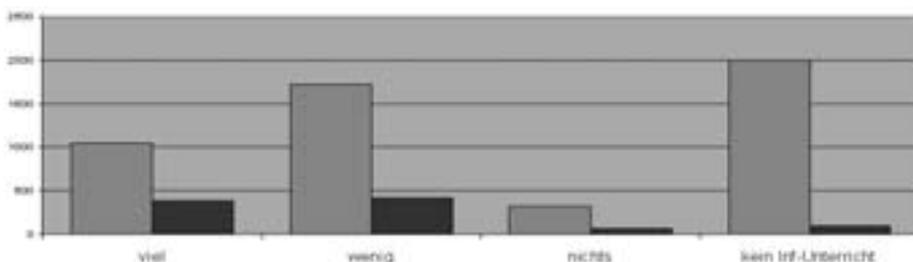


Abbildung 4: Ich habe im Informatikunterricht der Sekundarstufe ... gelernt

Die Hypothese, dass für viele SchülerInnen das Informatikangebot (noch immer) ein Grund ist, in der Schule zu verbleiben oder - für HauptschülerInnen - ein Oberstufengymnasium zu wählen, wurde nicht bestätigt. Nur jeder Achte gab an, dass das IT/Informatikangebot der zur Wahl stehenden weiterführenden Schule (Sekundarstufe II) für diese Entscheidung ausschlaggebend war. Diese Erkenntnis korreliert gut mit dem derzeit zu beobachtenden Trend, dass eine verstärkte IT/Informatik-Profilbildung nicht mehr als das werbeträchtige „Lockmittel“ für die in großer Konkurrenz stehenden Schulen bezeichnet werden kann, wie es noch am Beginn der Autonomiewelle in den 90-er Jahren war.

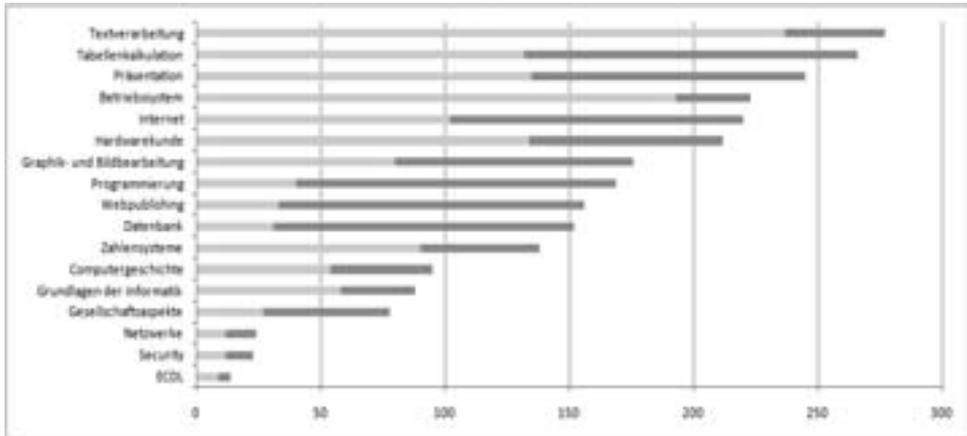


Abbildung 6: Inhalte 9. Schulstufe aus Lehrersicht (n = 408), 1. und 2. Semester

Die Themenstreuung und die Stoffgebiete sehen aus der Perspektive der Informatik-Lehrenden etwas anders aus als von den SchülerInnen wahrgenommen. Die Verteilung der Themen auf das 1. und 2. Semester (Abb. 6) wird durch zwei Momentaufnahmen, die von den SchülerInnen der 9. Schulstufe zum Lehrstoff zu Beginn des 1. Semesters und mitten im 2. Semester (unmittelbar vor oder nach Osterferien) bestätigt.



Abbildung 7: Dominierende Inhalte zu Beginn am Beginn der 9. Jahrgangsstufe

Die SchülerInnen der 9. Schulstufe konnten im Fragebogen aus vordefinierten Themen jenes Gebiet wählen, das zu Beginn des 1. Semesters unterrichtet wurde. Mehr als zwei Drittel der 7141 antwortenden SchülerInnen hatten keine Probleme sich zu erinnern. Was in Abb. 7 auffällt ist, dass viele SchülerInnen den Unterschied zwischen Softwarewerkzeug, Produktbezeichnung und Markenname nicht unterscheiden.



Abbildung 8: Stoffgebiete vor und nach den Osterferien

Offenbar dominiert im Unterricht das Lehrplankapitel „Beherrschung von Standardsoftware“, zumal aus der Sekundarstufe I auf keine gesicherten Kompetenzen aufgebaut werden kann. Ca. 60% der Informatiklehrenden sehen darin allerdings kein großes Problem und gaben an, den „digital gap“ zu nutzen, in dem informatisch vorgebildete SchülerInnen im Sinne des sozialen Lernens tutorielle Funktionen übernehmen. Andererseits haben sich zwei Drittel der Informatiklehrenden für zentral empfohlene Bildungsstandards aus Informatik für die 14-jährigen SchülerInnen am Ende der Sekundarstufe I ausgesprochen, um in der 9. Schulstufe von einer einigermaßen einheitlichen informatischen Vorbildung der SchülerInnen ausgehen zu können. Weitere, hier nicht angeführte Detailanalysen ergeben ein extrem variantenreiches und heterogenes Bild des Informatikunterrichts mit einer großen Bandbreite an speziellen Themen und spezifischen Softwareprodukten, die im unterschiedlichen Fach- und Werkzeugwissen der Informatiklehrenden begründet ist.

Informatikunterricht jenseits der dominierenden Office-Bedienschulung findet statt, allerdings in einem kleineren Umfang. Das Thema Programmierung wird, obwohl im Lehrplan vage mit „Grundprinzipien von Automaten, Algorithmen und Programmen kennen lernen“ beschrieben, von knapp weniger als der Hälfte der InformatiklehrerInnen behandelt. Ebenso verhält es sich mit den Themenbereichen Webdesign und Datenbanken. Daher scheint es wenig überraschend, dass sogar aus Sicht der SchülerInnen der 10. Schulstufe das in der 9. Schulstufe wahrgenommene „Bild der Informatik“ unvollständig scheint. Ob dies mit ein Grund dafür ist, dass die Begeisterung für das Wahlpflichtfach Informatik mit ca. 20% eines Jahrganges nicht mehr so groß ist wie noch vor einigen Jahren mit einem Anteil von einem Drittel, muss erst näher analysiert werden. Noch immer ist das Wahlpflichtfach Informatik männlich dominiert. Unter den Jugendlichen der 9. Schulstufe, die sich für dieses Wahlpflichtfach interessieren, kommen auf ein Mädchen vier Burschen.

5.4 Informatikunterricht in der 10. - 12. Schulstufe

Von einer Harmonisierung und Standardisierung des Informatikunterrichts ab der 10. Schulstufe kann derzeit keine Rede sein. Die Schulen bestimmen weitestgehend autonom, was und wie im Informatikunterricht der 16 – 18 Jährigen unterrichtet wird, zumal der Lehrplan für diese Schulstufen als wahllose Anordnung von Themenbereichen (siehe Kap. 3) einen großen Gestaltungsspielraum lässt, der von der Mehrheit der Informatiklehrenden akzeptiert wird. Eine Standardisierung für die (mündliche) Informatik-Matura wird mehrheitlich abgelehnt. Die Vielfalt der Themen und Werkzeuge im Rahmen der Fortbildungswünsche der Lehrerschaft macht diese Ablehnung verständlich.



Abbildung 9: Fortbildungswünsche von Informatiklehrenden

Über das Wahlpflichtfach hinaus gibt es, ähnlich der Sekundarstufe I, auch in der Sekundarstufe II in mittlerweile einem Drittel aller Gymnasien (Stand 2004) vom Schulstandort abhängige informationstechnologische Schwerpunktbildungen mit diversen Organisationsformen und schulautonomen Lehrplänen. Diese Profilbildungen, Kennzeichen dezentraler Bildungssteuerung, an den einzelnen Schulstandorten führen derzeit zu einer (in Kauf genommenen) Unübersichtlichkeit. Der Informatikunterricht sowie IT-relevante Fächer und Fachkombinationen sind davon in hohem Maße betroffen.

6 Fazit

Als Resümee bleibt festzuhalten: Der Informatikunterricht an den Gymnasien Österreichs kann je nach subjektiver Sichtweise entweder als eine gewünschte, bunt blühende Wiese oder aber als ein nicht zu tolerierendes Patchwork, dem klare Strukturen fehlen, bezeichnet werden. Die den Informatikunterricht bestimmenden Parameter wie infrastrukturelle Rahmenbedingungen, InformatiklehrerInnen, bundesweit unkoordinierte Fortbildungsveranstaltungen, offene Lehrpläne, das Fehlen von Bildungsstandards und eine nicht vorhandene Lehrbuchkultur weisen an den AHS Österreichs auf eine beispiellose Heterogenität und wohl auch Unverbindlichkeit hin. Im Rahmen einer wissenschaftlich fundierten empirischen Studie wurde die derzeitige Situation lediglich beschrieben, eingeschätzt und interpretiert. Den schulpolitischen Entscheidungsträgern ist es vorbehalten, daraus Schlüsse zu ziehen und Maßnahmen für eine Konsolidierung und Harmonisierung einzuleiten. Unter den momentanen Rahmenbedingungen, die den Schulen viel Autonomie einräumen, ist dieses Unterfangen nicht einfach. Wenn es allerdings gelingt, die Bildungsstandards im österreichischen Schulwesen auf informatische Bildung zu erweitern, und wenn endlich auch eine längst fällige und notwendige inhaltliche Bildungsreform geführt wird, stehen die Chancen gut, die Informatik in Allgemeinbildenden (Höheren) Schulen besser zu verankern.

Literatur und Referenzen

- [BD95] Börtz, J., Döring, N.: Forschungsmethoden und Evaluation für Sozialwissenschaftler. 2. Auflage, Berlin, Springer, 1995.
- [Ca05] Caba H. In: Micheuz P. (Hrsg.): CDA-Sonderheft 5/2005, CDA-Verlag, Linz, 2005
- [Gr99] Greig A., Taylor J.: Doing research with children. London, Sage, 1999
- [Ha97] Haider G.: Indikatoren zum Bildungssystem, Studienverlag, Innsbruck, 1997, S. 142
- [LP04] Informatik-Lehrplan AHS-Oberstufe Österreich (geprüft am 20.2.2005)
URL: http://www.bmbwk.gv.at/medienpool/11866/lp_neu_ahs_14.pdf
- [LP99] Lehrplan 2000 Unterstufe, <http://www.gemeinsamlernen.at> (geprüft am 15.2.2009)
- [Mi03] Mittermeir R., Schulinformatik – ein Fach oder ein Gebiet? In: [Re03], S. 7
- [Re03] Reiter A., Schulinformatik in Österreich, Ueberreuter-Verlag, Wien, S. 150

Softwaretools für kreatives Lernen im Informatikunterricht

Ralf Romeike

Didaktik der Informatik
Universität Potsdam
A.-Bebel-Str. 89
14482 Potsdam
romeike@cs.uni-potsdam.de

Abstract: Wohl kein Schulfach wird so maßgeblich durch seine Werkzeuge geprägt wie die Informatik. In diesem Artikel wird das Potential von Softwaretools zur Förderung kreativen Lernens im Informatikunterricht untersucht. Hierzu werden Kriterien beschrieben, die für Tools mit diesem Ziel gelten sollten und auf traditionelle und neue im Informatikunterricht verwendete Tools angewandt. Während traditionelle Tools im Informatikunterricht kreatives Lernen bisher kaum fördern, stellen v. a. neue Entwicklungen eine gute Arbeitsumgebung für Kreativität im Informatikunterricht bereit.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21. - 24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Spielend Sortieren mit Las Vegas Cardsort

Christian Wach
Dr.-Ing. Jens Gallenbacher

Technische Universität Darmstadt
Didaktik der Informatik
Hochschulstr. 10
64289 Darmstadt

wach@cs.tu-darmstadt.de
gallenba@cs.tu-darmstadt.de

Abstract: Las Vegas Cardsort ist ein einfaches Lernspiel, in dem die Spieler Karten sortieren. Las Vegas Cardsort wird von Schülern nicht als ein Lernprogramm, sondern als ein Minispiel wahrgenommen, dementsprechend hoch ist die Motivation und die Bereitschaft, mit Las Vegas Cardsort zu arbeiten. Es ergeben sich vielfältige Einsatzmöglichkeiten für den Unterricht. Las Vegas Cardsort kann auch ohne Programmierkenntnisse an den eigenen Unterricht angepasst werden.

1 Einleitung

Informatikunterricht im Jahr 1984, dem ersten INFOS-Jahr, stand zumeist voll unter dem Motto, einfache Vorgänge möglichst effizient in winzigem Speicherplatz unterzubringen. Das Sortieren war da ein beliebtes Unterrichtsbeispiel - mit Bubblesort als Pflicht und Quicksort als Kür. Heute, 25 Jahre später, präsentieren wir nun wieder das Sortieren von Spielkarten im Unterricht. Wo bleibt die Entwicklung?

Mit den GI-Bildungsstandards haben wir einen großen Schritt unternommen, dem Informatikunterricht eine solide Grundlage zu geben. Schon die zugehörigen Beispiele zeigen aber, dass Sortieren nach wie vor eine wichtige Rolle in der Vermittlung hat - vor allem für die Inhaltskompetenz Algorithmik. Sortieren ist einerseits alltäglich: Jede Schülerin und jeder Schüler hat bereits einen antrainierten Algorithmus dafür parat, den es zu erkunden und zu notieren gilt.

Das Sortieren ist andererseits sehr einfach! Gelingt es, den antrainierten Algorithmus außer Kraft zu setzen, lässt sich sehr schnell von neuem ein Lösungsansatz finden und erproben. Besonders spannend ist hier die Phase, in der sich im Kopf eine neue Handlungsstrategie formt und festigt.

Las Vegas Cardsort ist ein Spiel, in dem die Spieler Karten sortieren und dabei bestimmte Regeln einhalten müssen - mehr nicht. Trotzdem - oder vielleicht auch gerade deshalb - ist es eine sehr motivierende Plattform, vor der Kinder und Jugendliche viel Zeit verbringen

können, um ihre Sortierkünste und damit ihren Highscore zu verbessern. Dabei festigt sich über die Zeit immer mehr eine bestimmte - von Spieler zu Spieler unterschiedliche - Handlungsstrategie. Es ergeben sich vielfältige Einsatzmöglichkeiten im Unterricht.

Erfolgreich war Las Vegas Cardsort bereits im Rahmen der Ausstellung „Abenteuer Informatik“ (s. Abb. 1), die inzwischen bereits vielfach an unterschiedlichen Orten von sehr vielen Kindern und Jugendlichen, teilweise auch im Klassenverband, besucht wurde.

Las Vegas Cardsort ist unter der GNU General Public License 2 lizenziert und kann kostenlos bezogen werden[KMW⁺].

Abbildung 1: Kind beim Spielen von Las Vegas Vardsort an einem interaktiven Whiteboard



2 Las Vegas Cardsort

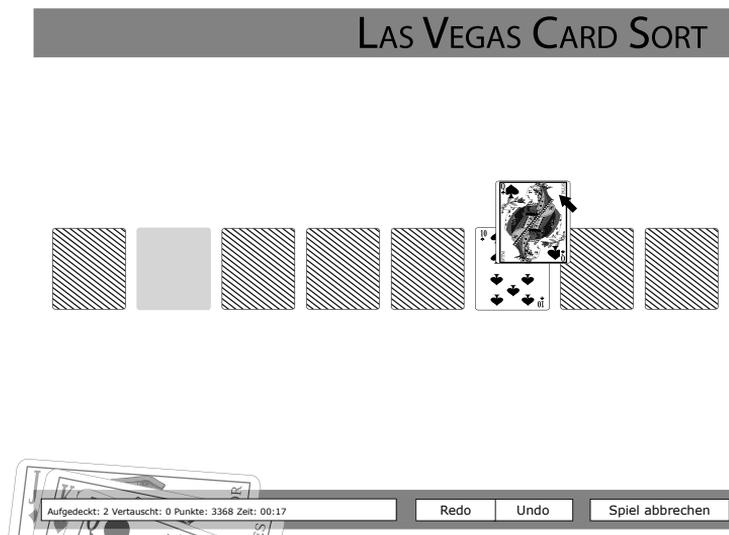
Las Vegas Cardsort wurde an der Fachdidaktik der Informatik der Technischen Universität Darmstadt von Studierenden im Rahmen eines Praktikums implementiert. Ziel der Entwicklung war ein System, das Schülern unterschiedlicher Jahrgangsstufen einen intuitiven und handlungsorientierten Zugang in das Themenfeld Sortieralgorithmen bietet. Las Vegas Cardsort ist eine reduzierte Algorithmen-Simulation[Fal01], die sich dem Schüler als Minispiel präsentiert.

Schüler kennen Sortieraufgaben bereits aus ihrem Alltag. Bei zahlreichen Kartenspielen ist

es für den Spieler hilfreich, die Spielkarten auf der Hand gemäß einer bestimmten Ordnung zu sortieren. Bei der Sortierung von Spielkarten befolgen Schüler in den seltensten Fällen bewusst eine konkrete Handlungsvorschrift, sondern ordnen die Karten intuitiv um. Las Vegas Cardsort knüpft an die Alltagserfahrung der Schüler an, und soll zum Nachdenken über die verwendete Strategie anregen.

Las Vegas Cardsort präsentiert sich Schülern als optisch ansprechendes Spiel, in dem Spielkarten sortiert werden müssen. Dem Spieler werden mehrere Spielfelder angeboten, in denen verschiedene Sortieraufgaben bewältigt werden können. Die Spieloberfläche eines solchen Spielfeldes ist in Abb. 2 abgebildet. Die Spielkarten liegen verdeckt auf der Spielfläche und müssen zunächst aufgedeckt werden. Je nach Spielfeld werden Karten auch wieder automatisch verdeckt und können miteinander vertauscht oder an einer anderen Stelle eingefügt werden. Das Spiel endet, wenn alle Karten korrekt sortiert sind.

Abbildung 2: Die Spieloberfläche von Las Vegas Cardsort

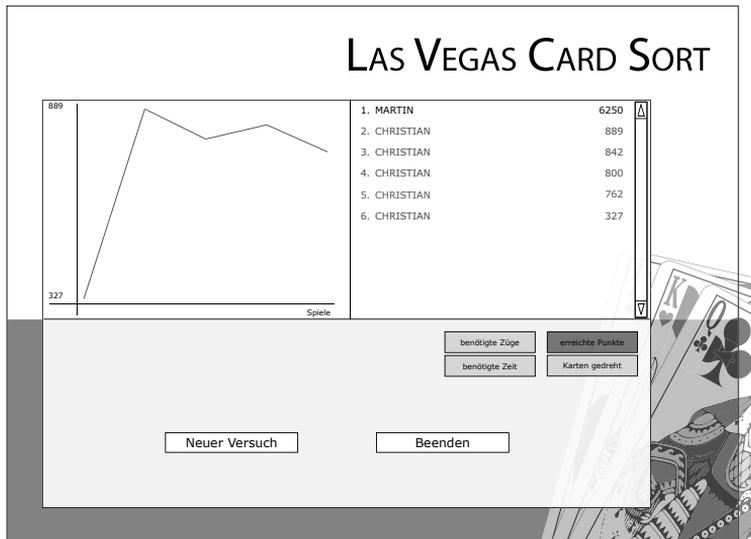


Jedes erfolgreiche Spiel wird mit einer Punktzahl bewertet. Diese Punktzahl wird aus der Anzahl benötigter Operationen (Aufdecken, vertauschen, ...) und der Zeit berechnet. Die erreichte Punktzahl wird im Profil gespeichert und kann mit den Leistungen der Mitschüler in der Highscoretabelle verglichen werden (s. Abb. 3). Wenn vom Lehrer eine Datenbank bereitgestellt wird, können die Punkte auch rechnerübergreifend verglichen werden.

Die Benutzerführung ist bewusst einfach gehalten. Las Vegas Cardsort wird ausschließlich mit der Maus bedient und kann so auch ohne weiteres mit Tablet PC oder interaktivem Whiteboard genutzt werden. Sobald der Mauszeiger eine Karte berührt, wird diese aufgedeckt. Vertauschungen von Karten werden mit Drag und Drop durchgeführt.

Las Vegas Cardsort besteht aus mehreren Spielfeldern, in denen verschiedene Sortier-

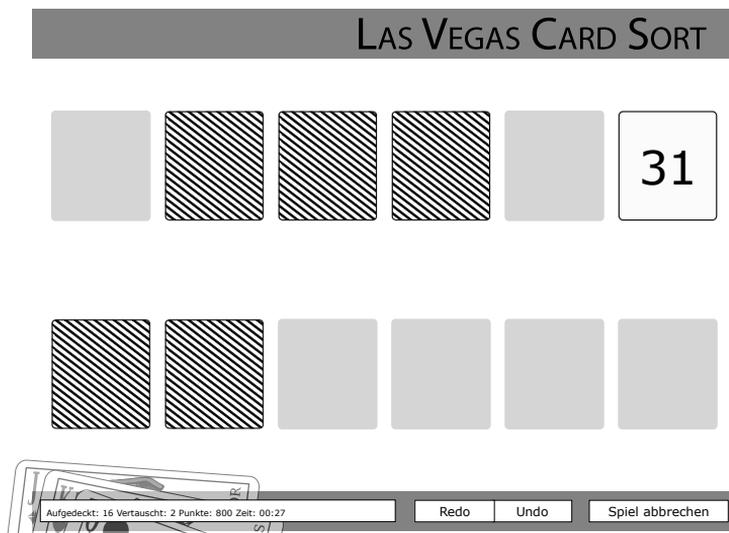
Abbildung 3: Die Highscore- und Statistikansicht von Las Vegas Cardsort



aufgaben bewältigt werden können. Unterschiede ergeben sich nur aus den Nutzereinschränkungen bei der Sortierung und den zu Grunde liegenden Datenstrukturen. So ist beim Spielfeld *Freies Spiel* ein Feld gegeben, das in-place sortiert werden soll. Es ist dem Schüler erlaubt beliebige Karten aufzudecken, miteinander zu vertauschen und sie an einer anderen Stelle einzusortieren, so dass die übrigen Karten entsprechend verschoben werden. Variationen dieses Spielfeldes sind die Spielfelder *Nur Tausch* und *Zwei Felder*. In der Variante *Nur Tausch* kann jede Karte aufgedeckt werden, allerdings können Karten nur paarweise miteinander vertauscht werden. Das Einsortieren einer Karte ist nicht mehr möglich. In der Variante *Zwei Felder* müssen die Karten des Eingabefeldes in das Ausgabefeld sortiert werden (not in-place Sortierung, s. Abb. 4).

Jedes Spielfeld kann mit unterschiedlichen Kartensätzen gespielt werden. Der Nutzer kann zwischen Spielkarten (Pik, 2-As), Karten mit Zahlen (1-99) und Karten mit Buchstaben wählen. Für jeden Kartensatz sind darüber hinaus die Verteilung der Karten, sowie die Vorsortierung wählbar. Las Vegas Cardsort bietet drei unterschiedliche Vorsortierungen der Karten: *Standard*, *Rückwärts* und *Zufällig*. Die Standardsortierung von Las Vegas Cardsort ordnet höher- und niederwertige Karten alternierend im Feld an. Mit der Standardsortierung werden zufällige Folgen erzeugt, die einen ähnlichen Grad der Vorsortierung aufweisen. Würde die Folge nur rein zufällig erzeugt, wären die Ergebnisse zweier Spiele nicht vergleichbar, da der Grad der Vorsortierung einen sehr deutlichen Einfluss auf die zu erreichenden Punkte hat.

Abbildung 4: Das Spielfeld *Zwei Felder*; Sortiere Karten von oben nach unten



3 Anwendungen im Unterricht

Las Vegas Cardsort wurde vor allem für den Schulunterricht der Sekundarstufe I konzipiert. Wenn die vom Arbeitskreis Bildungsstandards vorgeschlagenen Standards[P⁺08] Einzug in die Lehrpläne der Bundesländer finden, werden Lehrkräfte in gesteigertem Umfang Fachinhalte auf unterschiedlichen Niveaus unterrichten müssen. Im Inhaltsbereich Algorithmen existieren hier bereits ausgearbeitete Konzepte, die zum Beispiel ein Robotersystem verwenden [FK][BFRW06, S. 109ff.]. Es ist ein großer Vorteil solcher Systeme, dass Aufgaben sowohl mit Papier und Stift, als auch durch Programmierung in einer einfachen und reduzierten Programmiersprache gelöst werden können. Allerdings erfordern diese Systeme eine gewisse Einarbeitungszeit, bevor Schüler in der Lage sind konkrete Handlungsvorschriften zu formulieren.

Das Sortieren von Spielkarten setzt unmittelbar an den Alltagserfahrungen der Schüler an. Schüler sind sofort in der Lage, eine Menge von Spielkarten zu sortieren. Las Vegas Cardsort versetzt die Schüler in eine Wettbewerbssituation: Welcher Schüler sortiert n Karten am schnellsten bzw. mit der höchsten Punktzahl? In dieser Situation ist die verwendete Handlungsvorschrift zur Sortierung von besonderem Interesse. Die Frage nach einer erfolgreichen Handlungsvorschrift, bzw. Spielstrategie, ergibt sich im Unterrichtsgeschehen von selbst. Schüler, die einen hohen Punktwert erzielen, stellen ihre Strategie anderen Schülern vor, die ihrerseits selbst ein besseres Spielergebnis erzielen möchten. Die präsentierten Handlungsvorschriften sind zu diesem Zeitpunkt eher vage und werden den Anforderungen an Algorithmen nicht gerecht.

Im Unterricht sollen die Spielstrategien der Schüler schrittweise algorithmisiert werden. Denkbar sind sprachliche Handlungsvorschriften und die Darstellung in Diagrammen. Ausgeführt werden die so formulierten Handlungsvorschriften von Mitschülern in Las Vegas Cardsort. Nach wenigen Iterationen sind diese Handlungsvorschriften idealerweise soweit formalisiert, dass diese durch Mitschüler ausgeführt werden können. Automatisierbar sind die so entstandenen Handlungsvorschriften zu diesem Zeitpunkt jedoch noch nicht. Zur Abarbeitung ist es erforderlich, dass sich der ausführende Schüler mehrere Dinge, insbesondere Positionen im Feld, merkt. Die Formalisierung dieser Gedächtnisnutzung gestaltet sich ohne weitere Hilfsmittel schwierig. Las Vegas Cardsort bietet mit Markern ein eingeschränktes Variablenkonzept. Ein Marker ist ein farbiger Spielstein, mit dem einzelne Felder markiert werden können. Die Karten auf markierten Feldern sind immer aufgedeckt.

Mit Las Vegas Cardsort werden die Spielfelder *Mit Spielsteinen* und *3 Spielsteine* (s. Abb. 5) mitgeliefert. Beide Spielfelder enthalten jeweils drei Marker und unterscheiden sich nur im verwendeten Regelwerk. So können in Spiel *3 Spielsteine* Karten nur durch Marker aufgedeckt werden und es ist dem Spieler erlaubt markierte Karten miteinander zu tauschen. Der Tausch von nicht markierten Karten ist nicht möglich. In Spiel *Mit Spielsteinen* sind die Marker nur eine Information für den Spieler und schränken die Zugmöglichkeiten im Spiel nicht weiter ein. Durch die Verwendung von drei Markern können die gängigen quadratischen Sortierverfahren algorithmisch formuliert werden. Als Beispiel ist in Abb. 5 Selectionsort in einem Nassi-Shneiderman-Diagramm dargestellt.

3.1 Las Vegas Cardsort im Informatikunterricht der Oberstufe

Las Vegas Cardsort, vor allem für den Einsatz in der Sekundarstufe I entwickelt, kann auch im Unterricht der Sekundarstufe II sinnvoll eingesetzt werden. Die elementaren Sortierverfahren sind in Hessen schon für den Grundkurs verpflichtender Bestandteil des Unterrichtes [Hes, S. 14]. Sortierverfahren in $O(n \log n)$ sind im Grundkurs nur noch fakultativ vorgesehen, im Leistungskurs allerdings obligatorisch.

Tournamentsort [Gal08, S. 62ff.] gilt als besonders einfacher $O(n \log n)$ Algorithmus. Analog zu einem Turnier treten jeweils zwei Karten gegeneinander an. Die kleinere Karte gewinnt und kämpft auf der nächst höheren Turnierstufe weiter. So entsteht ein Turnierbaum, in dessen Wurzel die kleinste Karte gespielt wird. Die jeweils beste Karte wird dem Turnierbaum entnommen und in das nächste freie Feld eines Zielfeldes gelegt. Der Turnierbaum wird in jeder Iteration nicht vollständig neu berechnet, da die vorherigen Spielergebnisse wieder verwendet werden können.

Tournamentsort ist Schülern auf ikonischer oder enaktiver Ebene leicht zugänglich. Hier bietet Las Vegas Cardsort mit dem Spielfeld Tournamentsort (s. Abb. 6) einen solchen Zugang. Selbst die Betrachtung der Laufzeit des Algorithmus kann auf ikonischer Ebene erfolgen. Hierzu muss die Anzahl der Knoten im Baum ($2n$), sowie die Höhe des Baumes ($\log n$) bestimmt werden. Da es in diesem Fall genügt, nur vollständige Bäume zu betrachten, gelingen diese Abschätzungen sehr einfach.

Abbildung 5: Spielfeld 3 Spielsteine; Selectionsort mit 3 Markern

SELECTIONSORT

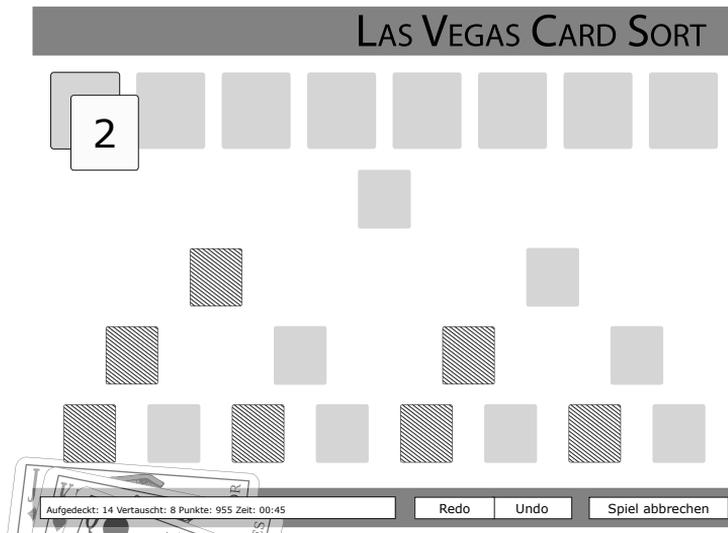
Lege BLAU auf Feld 1
Lege GRÜN auf Feld 1
LEGE ROT auf Feld 1
Solange BLAU nicht auf letztem Feld liegt
Solange ROT nicht auf letztem Feld liegt
Bewege ROT 1 Feld nach rechts
WENN Karte ROT kleiner als Karte GRÜN
Wahr Falsch
Lege GRÜN auf ROT ∅
Tausche Karte BLAU mit Karte GRÜN
Bewege BLAU 1 Feld nach rechts
Lege GRÜN auf BLAU
Lege ROT auf BLAU

Deutlich schwieriger gestaltet sich die Implementierung von Tournamentsort, da eine geeignete Modellierung von binären Bäumen benötigt wird. Wenn binäre Bäume nicht weiter im Unterricht behandelt werden sollen – denn oft sind diese ja nur fakultativ zu unterrichten – kann durch den Lehrer die sequentielle Darstellung in Arrays bereitgestellt werden. Sollen Bäume weiterhin Gegenstand des Unterrichts sein, empfiehlt sich eine verkettete Modellierung. Soll im Unterricht nur ein Ausblick auf effiziente Sortieralgorithmen gegeben werden, kann auf die Implementierung durch die Schüler verzichtet werden. Betrachtungen zu Korrektheit und Komplexität können in diesem Fall durch den Schüler auf der ikonischen Ebene erfolgen.

4 Anpassung von Las Vegas Cardsort

Ein wichtiges Entwicklungsziel von Las Vegas Cardsort war die Adaptierbarkeit des Programms durch Lehrkräfte an die Anforderungen im Unterricht. Die Konfiguration von Las Vegas Cardsort kann ohne Programmierkenntnisse mit XML Dateien an die Wünsche des Benutzers angepasst werden. Im Auslieferungszustand sind alle Spielfelder, Kartensets, Verteilungen und Vorsortierungen in Las Vegas Cardsort aktiviert. Falls den Schülern Las

Abbildung 6: Das Spielfeld Tournamentsort; Der Turnierbaum (unten); Das Zielfeld(oben)



Vegas Cardsort nur in einem reduzierten Umfang dargeboten werden soll, können einzelne Komponenten in der zentralen Konfigurationsdatei einfach deaktiviert werden [HKMS, S. 45ff.].

Zur Speicherung der Benutzerprofile und Highscoretabellen wird das Persistenzframework Hibernate genutzt, das eine eingebettete HSQLDB Datenbank nutzt. Soll eine gemeinsame Datenbank genutzt werden, kann diese ebenfalls angebunden werden [HKMS, S. 45ff.].

Las Vegas Cardsort ist Open Source Software ¹ und kann an definierten Schnittstellen erweitert werden [HKMS, S. 47ff.]. Eine Erweiterung im Quelltext ist allerdings nur nötig, wenn neue Komponenten hinzugefügt werden sollen. Neue Spielfelder lassen sich unter Verwendung bereits bestehender Komponenten ohne Quelltextmodifikation neu erzeugen oder aus vorhandenen Spielfeldern ableiten.

Spielfelder sind in Las Vegas Cardsort nicht programmiert, sondern werden lediglich in einer formalen Spezifikation definiert. So können auch Lehrkräfte, die nicht über Erfahrungen in der Softwareentwicklung mit Java verfügen, Las Vegas Cardsort mit wenig Aufwand an die Erfordernisse des eigenen Unterrichts anpassen. Im folgenden Abschnitt soll diese Erweiterbarkeit exemplarisch illustriert werden.

¹as Vegas Cardsort ist unter der GNU General Public License 2 (GPL v. 2) lizenziert

4.1 Anpassung und Erstellung von Spielfeldern

Spielfelder werden in Las Vegas Cardsort per XML Spezifikation (s. Abb. 7) erzeugt. Dort werden bereitgestellte Komponenten konfiguriert und zusammengestellt. In Abb. 7 sind die wesentlichen Bestandteile einer Spielfelddefinition aufgeführt. Ein Spielfeld besteht aus:

Abbildung 7: Aufbau einer Spielfelddefinition

```
<?xml version="1.0" encoding="utf-8"?>
<lvcs:gameDefinition
  xmlns:lvcs="http://las-vegas-cardsort.org/game"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <lvcs:metaData>
    <!-- Definition der Kopfdaten -->
  </lvcs:metaData>
  <lvcs:dataStructures width="n" height="m">
    <!-- Definition der Datenstrukturen -->
  </lvcs:dataStructures>
  <lvcs:markers>
    <!-- Definition der Marker -->
  </lvcs:markers>
  <lvcs:initialize>
    <!-- Initialisierung -->
  </lvcs:initialize>
  <lvcs:ruleSets>
    <!-- Definition der Regeln -->
  </lvcs:ruleSets>
  <lvcs:validEndState>
    <!-- Definition der gueltigen Endzustaende -->
  </lvcs:validEndState>
</lvcs:gameDefinition>
```

In den *Kopfdaten* werden externe und interne Bezeichner für das Spielfeld, Autor, Versionsnummer und Bewertungsfunktion definiert. Bewertungsfunktionen bestimmen aus der Anzahl der Karten im Spiel, der benötigten Zeit und der Operationsanzahl eine Punktzahl.

Im Bereich *Datenstrukturen* werden die zu verwendenden Datenstrukturen erzeugt und auf dem Spielfeld angeordnet. Die Datenstrukturen Liste und Baum sind bereits definiert. Jede Datenstruktur erhält einen eindeutigen Bezeichner. Die Spielfläche eines Spielfeldes wird als beliebiges äquidistantes $n \times m$ Gitter angesprochen. Auf diesem Gitter können dann Datenstrukturen unter Angabe von Position, Breite und Höhe positioniert werden.

Regeln, die die möglichen Operationen auf Datenstrukturen einschränken, können im entsprechenden Bereich (s. Abb. 7) definiert werden. Sind keine Regeln angegeben können beliebige Karten aufgedeckt und vertauscht werden. *Endzustände* definieren die möglichen Spielzustände, bei denen eine Partie erfolgreich gelöst wurde. In den meisten Fällen wird in diesem Bereich definiert in welcher Datenstruktur bei Spielende die Karten aufsteigend sortiert enthalten sein sollen.

Die Definition von Spielfeldern ist im Handbuch ausführlich beschrieben [HKMS, S. 13ff.].

An dieser Stelle sollen die Möglichkeiten zur Erstellung neuer Spielfelder an einem Beispiel illustriert werden. Bereits in Abschnitt 3 wurde das Spielfeld 3 *Spielsteine* im Kontext des Unterrichtes vorgestellt. In diesem Spielfeld werden eine Liste und drei Marker verwendet. Karten sollen nicht durch den Mauszeiger aufgedeckt werden und sind nur aufgedeckt, wenn sie markiert sind. Auch der Kartentausch unterliegt Einschränkungen. So dürfen nur markierte Karten miteinander vertauscht werden.

Die vollständige Beschreibung des gewünschten Spielfeldes ist in Abb. 8 Abgebildet. Zur Realisierung dieses Spielfeldes wird eine Liste als Datenstruktur genutzt. Da für dieses Spielfeld nur eine Datenstruktur benötigt wird, ist eine Unterteilung des Spielfeldes nicht nötig und die Liste *list1* kann auf das einzige Gitterfeld positioniert werden (Zeilen 14-18 in Abb. 8). Die drei benötigten Marker werden in den Farben rot, blau und grün angelegt (Zeilen 19-23). Die Liste *list1* nimmt die generierten Spielkarten in der erzeugten Reihenfolge auf (Zeile 25) und die drei Marker werden auf den ersten drei Feldern der Liste *list1* positioniert (Zeilen 26-28). Karten dürfen nur getauscht werden, wenn Start- und Zielfeld von einem Marker markiert sind (Zeilen 31-37). Das Aufdecken von Karten durch MouseOver ist nicht möglich (Zeile 38). Markierte Felder sind per Definition immer aufgedeckt. Das Spiel ist beendet, wenn Liste *list1* alle Karten in aufsteigend sortierter Reihenfolge enthält (Zeilen 40-44).

Auch komplexere Spielfelder lassen sich auf diese Weise einfach erstellen. In den meisten Fällen sollte eine Modifikation eines der mitgelieferten Spielfelder genügen, um den individuellen Anforderungen zu genügen.

5 Zusammenfassung und Ausblick

In der vorliegenden Fassung ist Las Vegas Cardsort für einen handlungsorientierten Zugang in das Themenfeld Sortieralgorithmen gut geeignet. Die Bedienung von Las Vegas Cardsort ist sehr einfach und erfordert keine umfangreiche Vorbereitung oder Einweisung. Las Vegas Cardsort wird von Schülern nicht als ein Lernprogramm, sondern als ein Minispiel wahrgenommen, dementsprechend hoch ist die Motivation und die Bereitschaft, mit Las Vegas Cardsort zu spielen.

Las Vegas Cardsort enthält bereits einige Spielfelder, die im Unterricht direkt verwendet werden können. Darüber hinaus können neue Spielfelder auch ohne Programmierkenntnisse in XML spezifiziert werden. Auch bestehende Spielfelder können so modifiziert und an den eigenen Unterricht angepasst werden.

Las Vegas Cardsort unterstützt das Protokollieren von allen Nutzeraktionen während des Spiels [HKMS, S. 33ff.] in XML Dateien. Die Protokollierung ist im Auslieferungszustand deaktiviert und muss explizit durch den Lehrer aktiviert werden. Wir hoffen mit diesen Daten Einblicke in die kognitive Entwicklung einer algorithmischen Problemlösung zu erhalten. Von besonderem Interesse sind hier die Veränderungen an der Spielstrategie im Laufe der Zeit und Anpassungen an geänderte Rahmenbedingungen.

So konnten wir bei bisherigen Testläufen Personen identifizieren, die in besonderem Maße mit heuristischen Strategien gearbeitet haben, die auf Annahmen der Kartenverteilung

Abbildung 8: Spezifikation des Spielfeldes 3 Spielsteine

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <lvcs:gameDefinition
03   xmlns:lvcs="http://las-vegas-cardsort.org/game"
04   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
05   <lvcs:metaData>
06     <lvcs:name>3 Spielsteine</lvcs:name>
07     <lvcs:gameID>lvcs_3marker_0_1_0x0fe3</lvcs:gameID>
08     <lvcs:revision>0.1</lvcs:revision>
09     <lvcs:author>Christian Wach</lvcs:author>
10     <lvcs:scoreExpression>
11       1000*(CARDS^2)/(MOVES+TIME+VIEWS)
12     </lvcs:scoreExpression>
13   </lvcs:metaData>
14   <lvcs:dataStructures width="1" height="1">
15     <lvcs:structure type="list" id="list1" scaleFactor="1">
16       <lvcs:presentation class="SimpleList" x="0" y="0" w="1" h="1"/>
17     </lvcs:structure>
18   </lvcs:dataStructures>
19   <lvcs:markers>
20     <lvcs:marker id="marker1" influencesPlaces="1" color="red"/>
21     <lvcs:marker id="marker2" influencesPlaces="1" color="blue"/>
22     <lvcs:marker id="marker3" influencesPlaces="1" color="green"/>
23   </lvcs:markers>
24   <lvcs:initialize>
25     <lvcs:fillStructure order="natural" target="list1" start="0"/>
26     <lvcs:setMarker markerID="marker1" target="list1.0"/>
27     <lvcs:setMarker markerID="marker2" target="list1.1"/>
28     <lvcs:setMarker markerID="marker3" target="list1.2"/>
29   </lvcs:initialize>
30   <lvcs:ruleSets>
31     <lvcs:ruleSet type="move" defaultPolicy="deny">
32       <lvcs:rule class="simpleDrag" type="allow"
33         source="structure.list1.*:influencedByMarker"/>
34       <lvcs:rule class="simpleDrop" type="allow"
35         source="structure.list1.*:influencedByMarker"
36         destination="structure.list1.*:influencedByMarker"/>
37     </lvcs:ruleSet>
38     <lvcs:ruleSet type="select" defaultPolicy="deny"/>
39   </lvcs:ruleSets>
40   <lvcs:validEndState>
41     <lvcs:endState class="checkStructure" assert="sorted"
42       target="list1" order="natural"/>
43   </lvcs:validEndState>
44 </lvcs:gameDefinition>
```

basierten. Solche Heuristiken können durch eine Veränderung der Verteilung der Kartenwerte deutlich gestört werden.

In der vorliegenden Version kann die Verteilung der Kartenwerte nur bei der Auswahl des Spielfeldes festgelegt werden. Eine Anpassung der Verteilungen zwischen konsekutiven Spielrunden ist nicht möglich. Unser Ziel ist es, unterschiedliche Problemlösestrategien maschinell zu identifizieren, so dass eine automatische Adaption an den Benutzer möglich wird. Hierfür benötigen wir allerdings Protokolldaten in ausreichendem Umfang. Daher möchten wir alle Lehrer, die Las Vegas Cardsort im Unterricht einsetzen, bitten, uns die Protokolldaten ihrer Klasse zur Verfügung zu stellen. Interessierte Lehrer können sich hierzu gerne an die Autoren wenden.

Literatur

- [BFRW06] Peter Brichzin, Ulrich Freiberger, Klaus Reinold und Albert Wiedemann. *Grundlagen der Informatik*. Oldenbourg Schulbuchverlag, 2006.
- [Fal01] Nils Faltn. Algorithmen lernen mit interaktiven Visualisierungen. In *INFOS '01: Informatikunterricht und Medienbildung, INFOS 2001, 9. GI-Fachtagung Informatik und Schule*, Seiten 87–96. GI, 2001.
- [FK] Ulli Freiberger und Ondrej Krsko. Robot Karol. <http://www.schule.bayern.de/karol/>.
- [Gal08] Jens Gallenbacher. *Abenteuer Informatik: IT zum Anfassen von Routenplaner bis Online-Banking*. Spektrum Akademischer Verlag, 2008.
- [Hes] Hessisches Kultusministerium. Lehrplan Informatik: Gymnasialer Bildungsgang Jahrgangsstufe 11 bis 13. <http://lernarchiv.bildung.hessen.de/lehrplaene/gymnasium/informatik/index.html>.
- [HKMS] Michael Herrmann, Thorsten Krause, Arno Mittelbach und Andreas Schmitt. Las Vegas Cardsort: Handbuch. <http://www2.di.informatik.tu-darmstadt.de/lvcs/handbuch>.
- [KMW⁺] Thorsten Krause, Arno Mittelbach, Christian Wach et al. Las Vegas Cardsort: Programmdownload. <http://www2.di.informatik.tu-darmstadt.de/lvcs/download>.
- [P⁺08] Hermann Puhlmann et al. Grundsätze und Standards für die Informatik in der Schule, 2008. <http://gi.informatikstandards.de/>.

Informatik im Kontext (IniK)

Ein integratives Unterrichtskonzept für den Informatikunterricht

Jochen Koubek

Universität Bayreuth
Sprach- und Literaturwiss. Fakultät
Digitale Medien
jochen.koubek@uni-bayreuth.de

Carsten Schulte

Freie Universität Berlin
Fachbereich Mathematik und Informatik
Didaktik der Informatik
Schulte@inf.fu-berlin.de

Peter Schulze

Landesinstitut für Schule und Medien
Berlin-Brandenburg (LISUM)
Abteilung II: Unterrichtsentwicklung
Peter.Schulze@lisum.berlin-brandenburg.de

Helmut Witten

Brandenburgische Str. 23
10707 Berlin
helmut@witten-berlin.de

Abstract: Kontextorientierter Unterricht ist in den Naturwissenschaften ein bekannter und erfolgversprechender Ansatz. Doch lohnt eine Übertragung auf den Informatikunterricht, der ja sowie so schon viel stärker Kontexte beachtet? Anwendungen, Auswirkungen und Projekte gehören hier zum Standard. Im Artikel werden wir ein Konzept für Informatik im Kontext (IniK) erläutern und den erhofften Mehrwert darlegen. Zentral ist der Entwurf eines einfachen Schemas zur Konstruktion kontextorientierten Unterrichts in der Informatik, der sich an Kontexten, Standards und Methodenvielfalt orientiert.

1 Einleitung

Die strenge Fächerung des Unterricht mit der inhaltlichen und personellen Ausrichtung an einer akademischen Fachdisziplin sorgt dafür, dass sich von Bildungsplänen über Lehrpläne bis zum Unterrichtsentwurf die jeweilige Gliederung des Bezugsfachs als tragende Kraft durchgesetzt hat, bei der bis auf wenige Ausnahmen Fachprinzipien und -logiken stärker betont werden als ihre Anwendung und Sichtbarmachung im Alltag. Die Orientierung an der Fachwissenschaft ist fraglos ein bedeutender Pfeiler moderner Schulpädagogik. Doch gilt es dabei, die Orientierung an einem sinnstiftenden Kontext nicht aus den Augen zu verlieren

Inbesondere in den so genannten MINT-Fächern (Mathematik, Informatik, Naturwissenschaften und Technik) wird immer wieder deutlich, dass oftmals eine Kluft zwischen fachlichen Lernzielen, der Strukturierung des Unterrichts und den Interessen, Erwartungen und der Motivation der Schülerinnen und Schüler auftritt.

Steht im Anfangsunterricht der Naturwissenschaften und der Technik der phänomenologische Zugang noch im Vordergrund, setzt sich in den folgenden Jahren die Fachsystematik immer weiter durch. Unter dem „Deckmantel“ der fachlichen Exaktheit werden die notwendigen Erklärungen immer komplexer und wissenschaftlicher, so dass die anfängliche Freude bald umschlägt. Das Hauptinteresse von jüngeren Schülerinnen und Schülern besteht im Wesentlichen im Bauen und Konstruieren, bei älteren überwiegt das Diskutieren und Bewerten. Jede Form von Berechnen, Symbolisieren und Formalisieren wird hingegen abgelehnt [EL06]. Deutlich spiegelt sich dieser Trend in der Wahl naturwissenschaftlich-technischer Fächer in der gymnasialen Oberstufe wider bzw. in der Entscheidung für ein Studium.

Für den naturwissenschaftlichen Unterricht gibt es deshalb bereits seit längerem verschiedene kontextorientierte Ansätze. Im Folgenden werden wir zunächst die verwandten Ansätze aus den Naturwissenschaften und der Informatik-Didaktik vorstellen, dann eine für die Informatik geeignete Definition der Kontextorientierung entwickeln, darauf aufbauend das Konzept IniK entwerfen und an einem Beispiel erläutern.

Das Projekt CHiK (Chemie im Kontext) will Schülerinnen und Schüler in „für sie bedeutsamen und nachvollziehbaren Zusammenhängen mit chemischen Fachinhalten“ konfrontieren; dabei sollen die gewählten Kontexte mehr als nur motivierende Aufhänger sein und den roten Faden der Unterrichtseinheit bilden ([FSR07], S. 274). Daher werden CHiK-Unterrichtseinheiten in vier Phasen gegliedert: 1) Begegnungsphase, in der sich die Schülerinnen und Schüler mit dem Kontext vertraut machen, 2) Neugier- und Planungsphase, in der Fragen formuliert werden, 3) Erarbeitungsphase, in der die Fragen beantwortet werden und 4) Vertiefungs- und Vernetzungsphase, in der die (fachlichen) Basiskonzepte im Mittelpunkt stehen und z.B. in andere Kontexte gestellt werden.

Im Projekt *piko* (Physik im Kontext) meint *Kontext* zum einen den Kontext, in dem ein fachlicher Inhalt eingebettet wird, um sinnstiftend zu sein, zum andern den Kontext, in dem gelernt wird ([MSD07], S. 266). In dem Projekt spielen neben der Orientierung an Kontexten weitere Aspekte eine Rolle, etwa Methodenvielfalt, die Förderung naturwissenschaftlichen Denkens oder auch das Einbeziehen moderner Technologien in die Themen des Unterrichts, um „die Bedeutung moderner Physik für innovative Technologien“ zu vermitteln – denn diese spielen eine immer wichtigere Rolle in der Lebenswelt (a. a. O., S. 268). Schließlich entstehen aus authentischen Erfahrungen mit Forschung und Arbeitswelt außerschulische Kontexte [IP07].

Das Projekt *bik* (Biologie im Kontext) versteht unter Kontexten, dass sie „Anknüpfungspunkte“ an die Alltagserfahrungen der Schülerinnen und Schüler liefern und den Kompetenzerwerb unterstützen. Kontexte sollen andererseits eine Brücke zu den wissenschaftlichen Anwendungsfeldern schlagen, müssen „dem Entwicklungsstand“ der

Lernenden angemessen sein und lebensweltliche oder gesellschaftliche Relevanz für die Schülerinnen und Schüler besitzen ([BA07], S. 282 f.). Zur Umsetzung werden in bik unter anderem Aufgaben entwickelt, die sich an fünf Kriterien orientieren: 1. Kontext der Aufgabe (lebensweltlich relevant?), 2. Basiskonzepte (welche fachlichen Konzepte werden benötigt?), 3. Kompetenzen (Welche Kompetenzen nach den Bildungsstandards sind erforderlich?), 4. Affektive Dimension (motivierend und interessant?), und 5. Unterrichtsphase (Funktion der Aufgabe, z. B. Lernaufgabe, Diagnoseaufgabe) ([BA07], S. 283).

Auch wenn die drei naturwissenschaftlichen Kontextprogramme durchaus ein unterschiedliches Verständnis von Kontextbezügen und ihrer Einbindung in den Unterricht haben, eint die drei Programme die Betonung der Bedeutung von lebensweltlichen Fragen der Schülerinnen und Schüler aus Alltagsbezügen, die für diese einen „sinnstiftenden“ Kontext (nach Muckenfuss, [MU95]) besitzen müssen.

Verwandte Ansätze in der Informatik-Didaktik

Das Konzept des „anwendungsorientierten Informatikunterrichts“ wurde in den 70er Jahren des letzten Jahrhunderts in Berlin (West) entwickelt: „Die Methoden und Verfahren der Informatik sollen im Kontext der Anwendungen der Datenverarbeitung in gesellschaftlichen Bereichen und deren Auswirkungen erlernt und angewandt werden, um somit die Schüler zum gesellschaftlich verantwortlichen Handeln zu befähigen“ ([SZ78], S. 47).

Die Anwendungsorientierung geht von den fachlichen Inhalten aus, die durch Kontexte angereichert werden. Die Kontextorientierung geht im Gegensatz dazu von den Kontexten aus, die den Unterricht strukturieren.

Was unter der Anreicherung mit Kontexten in der Anwendungsorientierung zu verstehen ist, wird an dem Beispiel einer Schülerdatei erläutert: „In diesem Projekt können die Schüler lernen, dass eine Schülerdatei nicht für verwaltungstechnische Zwecke an sich aufgebaut sein muss, wie z. B. zur Berechnung von Fehlstunden, sondern genau so für eigene Schülerinteressen entwickelt werden kann, wie etwas zur Ermittlung von Freistunden für Förderungsmaßnahmen oder für die Freizeitgestaltung“ (ebd.).

Ein zentraler Begriff des anwendungsorientierten Unterrichts war das „Betroffensein“ der Schülerinnen und Schüler: „Der Aspekt des Betroffenseins bestimmt den gesamten Informatikunterricht. Er prägt die Behandlung des Datenschutzes, die Darstellung und Auseinandersetzung mit EDV-Berufen [sowie] die Betrachtung der historischen Entwicklung“ ([GR76], S. 10).

Es ist strittig, inwieweit dieser anspruchsvolle Ansatz umgesetzt werden konnte (s. [EN04], S. 176 ff.). In jedem Fall erfuhr der Begriff der „Anwendungsorientierung“ im Zuge der Einführung der informationstechnischen Grundbildung (ITG) in den 80er Jahren einen Bedeutungswandel. In diesem Unterricht stand vielfach die Beschäftigung mit fertigen Computer-Anwendungen im Vordergrund, es sollte nicht mehr programmiert werden. Insofern entwickelten sich die „Anwendungen“ zu einem Synonym für Text-

verarbeitung, Tabellenkalkulation und Datenbank. Damit hat der Begriff der „Anwendungsorientierung“ für den Informatikunterricht alle Trennschärfe verloren.

Kontextorientierung kann ggf. auch leicht mit Projektorientierung verwechselt werden. Schließlich sollen Projekte den Alltags- und Lebensweltbezug der Lernenden berücksichtigen. Damit wird der Unterricht noch nicht unbedingt am Kontext ausgerichtet; Projekte stellen eine Unterrichtsmethode dar. Kontextorientierung kann in vielen Fällen projektartig umgesetzt werden, erschöpft sich jedoch nicht darin.

Einen umfassenden Überblick zu weiteren kontextorientierten Ansätzen in der Informatik-Didaktik (einschließlich seines eigenen) gibt Dieter Engbring in seiner Dissertation ([EN04]).

2 Informatik im Kontext

Informatik im Kontext basiert auf drei Prinzipien:

1. Orientierung an Kontexten
2. Orientierung an Standards für die Informatik in der Schule
3. Methodenvielfalt

2.1 Orientierung an Kontexten

Unterricht soll sich an der Lebenswelt der Schülerinnen und Schüler ausrichten. Dieser Wunsch ist nicht neu: *Situativer Unterricht*, *problemorientierter Unterricht*, *anwendungsorientierter Unterricht* oder *schülerzentrierter Unterricht* sind Schlagworte für vergleichbare Vorstellungen, die regelmäßig gefordert, im schulischen Alltag aber ebenso regelmäßig vergessen bzw. auf ein motivierendes Einstiegsproblem reduziert werden.

Kontextorientierter Unterricht richtet sich bei seinen didaktischen Entscheidungen zu Zielen, Inhalten, Methoden, Medieneinsatz, Verlauf etc. konsequent am Kontext aus. Was aber ist ein Kontext?

Zunächst einmal gehört jeder Kontext zu einer konkreten Situation, beide sind zusammen einmalig und unwiederholbar. Der **Kontext** einer Situation lässt sich als vieldimensionaler Handlungsrahmen betrachten. Jede Situation hat ihre technischen, sozialen, ethischen, rechtlichen, ökonomischen, ästhetischen etc. Aspekte, die zusammen den Kontext ausmachen. Kurzum: es gibt so viele Kontexte wie es Situationen gibt, jede Situation hat ihren eigenen vieldimensionalen Kontext [KK07].

Aber so wie verschiedene Situationen als ähnlich erlebt werden können, sind es auch ihre Kontexte. Der Philosoph Ludwig Wittgenstein hat für diesen Abstraktionsprozess den Begriff *Familienähnlichkeit* geprägt [WI53], durch den disparate aber ähnliche Phänomene zusammen geklammert werden, wobei die Ränder unscharf bleiben. Kontexte

sind demnach zueinander ähnlich, wenn sie in ihren Aspekten und ihren diskursiven Einordnungen (zum Diskursbegriff s. u.) ähnlich sind bzw. wenn von einigen dieser Aspekte abgesehen wird.

Insofern blickt kontextorientierter Unterricht nicht auf den singulären Kontext, sondern auf Unterscheidungsmerkmale von Kontextfamilien auf unterschiedlichem Abstraktionsniveau. Der Einfachheit halber sprechen wir im Weiteren dennoch von *Kontext* als *Menge von lebensweltlichen Themen bzw. Fragestellungen, die von den Schülerinnen und Schüler als zusammenhängend geordnet werden und die dadurch sinnstiftend auf deren Handlungen wirken*. Beispiele für derartige (abstrakten) Kontexte sind *Chatbots, Filesharing, Podcasts, Blogs, Soziale Netze* oder *RFID*. Sie fassen verschiedene Handlungssituationen und ihre (konkreten) Kontexte thematisch zusammen, die einander ähnlich sind und sich durch den im Kontexttitel ausgedrückten Oberbegriff von anderen Situationen wesentlich unterscheiden.

Während die Kontexte sich in aller Regel dem Erlebnishorizont des Einzelnen entziehen, stehen als Beobachtungsgegenstände ihre medialen Aufbereitungen zur Verfügung, die als Dokumente, Artikel, Kommentare, audiovisuelle Beiträge, Webseiten etc. zu Themen geordnet, diskutiert, ausgehandelt und vermittelt werden. Kontexte werden in ihrer Vieldimensionalität über diese Diskurse auch im Unterricht erschließbar. Der Begriff **Diskurs** wird hier verstanden als Äußerungszusammenhang zu einem bestimmten Thema (vgl. [KO05]). Beispiele für Diskurse, in denen Informatiksysteme und Jugendliche eine prominente Rolle spielen sind *Digitale Medien, Geistiges Eigentum, Datenschutz, Computerspiele, Soziale Netze, E-Commerce, Informationsfreiheit, Persönlichkeitsrechte, Künstliche Intelligenz (KI)*.

Informatik im Kontext fokussiert auf die lebensweltliche Einordnung von Informatiksystemen in diese Themen. Als zusätzliche Randbedingung im Rahmen des Schulunterrichts sollten die ausgewählten Kontexte Ereignisse aus dem Erfahrungshorizont der Schülerinnen und Schüler einschließen und somit einen Bezug zur Lebenswelt der Lernenden direkt erfahrbar machen. Zwar sollten Jugendliche sich auch mit Phänomenen jenseits ihrer Alltagserfahrung auseinandersetzen, aber nicht jeder gesellschaftlich relevante Kontext mit IT-Bezug eignet sich dafür, Jugendliche zu einer Bildungsbewegung zu motivieren.

Dekontextualisierung

Kontexte lassen sich unter vielfältigen Gesichtspunkten betrachten, beurteilen und verstehen. *Geistiges Eigentum* hat eine historische, rechtliche, politische, ökonomische, soziale, ethische, kulturelle und nicht zuletzt eine informationstechnische Dimension, die unabhängig voneinander behandelt werden können. In der Schule ist dies traditionell Aufgabe der verschiedenen Fachunterrichte. Unter **Dekontextualisierung** wird die Projektion des Kontexts auf die Basiskonzepte eines Fachgebiets verstanden. Ein Kontext würde in der Schule idealerweise im fächerverbindenden Unterricht von verschiedenen Seiten ausgeleuchtet. Auf diese Unterrichtsform sind aber weder die Lehrpläne noch die Unterrichtsmaterialien noch die Kollegien vorbereitet. Ein kontextorientierter Informatikunterricht hat damit verschiedene Ausgestaltungsoptionen, je nach fachdidaktischer

Sicht auf die Ziele und inhaltlichen Schwerpunkte, die Informatikunterricht in der Schule verfolgen sollte:

1. Im Informatikunterricht geht es um Informationstechnik und ihre mathematische Fundierung: Der Kontext wird hier lediglich als Aufhänger benutzt, als Motivationshilfe zum Einstieg in die Unterrichtseinheit. Von dort aus wird möglichst rasch zu den informationstechnischen Prinzipien abstrahiert. *Informatik im Kontext* bedeutet aus dieser Sicht die Freilegung der informationstechnischen und – mathematischen Anteile eines gegebenen Kontexts. Die Schüler lernen z.B. wie ein P2P-Netz auf Protokollebene funktioniert, nicht aber über die rechtliche und gesellschaftliche Bedeutung von File-Sharing-Netzen, was in dieser Sicht als Aufgabe des Gesellschaftskundeunterrichts gilt (oder des Ethikunterrichts oder des Rechtskundeunterrichts oder irgend eines anderen Unterrichts, nicht aber der Informatik).
2. Informatikunterricht behandelt primär Informationstechnik, jedoch mit gesellschaftlichen Aspekten. Dazu werden je nach Lehrer *Geschichte*, *Ökonomie*, *Recht* oder auch *Ethik* gezählt. *Informatik im Kontext* bedeutet hier, dass diesen Aspekte im Informatikunterricht Berechtigung eingeräumt wird. Entsprechend kann in verschiedene Richtungen dekontextualisiert werden, um ausgehend von dem gewählten Kontext technische, aber auch ethische, ökonomische oder historische Prinzipien freizulegen.
3. Informatik ist ein gesellschaftliches Fach mit verschiedenen, gleichberechtigten Aspekten. In dieser Extremposition sind Mathematik und Technik zwar wichtige aber nicht in jedem Fall die wichtigsten Betrachtungsweisen von Informatiksystemen. *Informatik im Kontext* bedeutet hier die Thematisierung verschiedener Dimensionen eines Kontextes, je nachdem, welcher Aspekt im Vordergrund stehen soll und vom Lehrer kompetent abgedeckt werden kann.

Die letzten beiden Positionen unterscheiden sich vor allem in ihrer Antwort auf Frage, welche gesellschaftlichen Dimensionen in welcher Gewichtung zur Informatik gehören. Gemeinsam ist ihnen, dass aus ihnen heraus versucht wird, fachübergreifend verschiedene Aspekte des Kontextes zu betrachten. Dieser Unterricht akzeptiert, dass die gesellschaftlichen Wechselwirkungen von Informationstechnik untrennbar zur Informatik dazu gehören und auch nicht getrennt von ihr zu behandeln sind.

Dekontextualisierung ist die Abstraktion vom flüchtigen Kontext zu den Grundprinzipien des Faches und den daraus abgeleiteten Kompetenzen (s. u.: Orientierung an Standards). Diese Bewegung ist dem Unterricht nicht fremd. Viele Unterrichtseinheiten werden ausgehend von einem Beispiel eingeführt, von dem anschließend die fachlichen Essenzen extrahiert und vertieft werden. Das Risiko liegt darin, dass dabei jene Teile der Klasse zurück gelassen werden, die in den Prinzipien das Problem nicht mehr wieder erkennen und die Bäume vor lauter Wald nicht mehr sehen. In diesem Fall wäre es Aufgabe des Lehrers, durch motivierende Zwischensequenzen den verloren geglaubten Bezug zur Lebenswelt wieder herzustellen. Hier liegen die Stärken des kontextorientierten Unterrichts, der qua Design sein Ausgangsproblem nicht vergisst und auch nach verschiedenen Dekontextualisierungen wieder zum ursprünglichen Kontext zurückkehrt.

Ebenso kann jede Unterrichtsform an lebensweltlichen Problemen wichtige Fachprinzipien entwickeln, ohne dabei die Welt aus den Augen zu verlieren, aber sie muss es nicht. Einfacher ist die Stufenfolge *Einstiegsproblem – Fachprinzipien – Fachprobleme*. Ein kontextorientierter Unterrichtsentwurf hingegen verbindet die Fachprinzipien eng mit einem Kontext, auf den regelmäßig Bezug genommen wird. Die Blickrichtung ist immer auch bottom-up von den Kontexten auf das Fach und nicht nur top-down auf das eine oder andere Beispiel – oder auch nicht, wenn die Zeit zum Jahresende hin wieder einmal knapp wird.

2.2 Orientierung an Standards für die Informatik in der Schule

Trotz der Orientierung an Kontexten soll der Unterricht über den Charakter des Exemplarischen hinaus Wissen und Fertigkeiten fördern, die auch in anderen Kontexten hilfreich sind. Kompetenzen werden verstanden als individuelle Disposition zur Lösung von Problemen in variablen Situationen (vgl. [WE01]).

Die Gesellschaft für Informatik e.V. hat am 24. Januar 2008 die vom GI-Arbeitskreis »Bildungsstandards« erarbeiteten *Grundsätze und Standards für die Informatik in der Schule* einstimmig als Empfehlung verabschiedet [GI08]. Bei der Formulierung der Grundsätze und Standards für den Informatikunterricht wurden „die Standards in fünf Inhalts- und fünf Prozessbereiche unterteilt. Das sind im Einzelnen als Inhaltsbereiche: Information und Daten [I1], Algorithmen [I2], Sprachen und Automaten [I3], Informatiksysteme [I4], Informatik, Mensch und Gesellschaft [I5]. Als Prozessbereiche sind dies: Modellieren und Implementieren [P1], Begründen und Bewerten [P2], Strukturieren und Vernetzen [P3], Kommunizieren und Kooperieren [P4], Darstellen und Interpretieren [P5]“ ([GI08], S. VI, Abkürzungen in eckigen Klammern von den Autoren).

Mit diesem Dokument liegt erstmalig eine Liste von Bildungsstandards für den Informatikunterricht vor, die Schülerinnen und Schüler nach Abschluss der Sekundarstufe I mindestens erreicht haben sollten. Dadurch ist es z. B. möglich, die Entwicklung von Lehrplänen und Lehrmaterialien bundesweit zu koordinieren und an einem gemeinsamen Verständnis von informatischer Schulbildung auszurichten.

Mit dieser Zielsetzung orientieren sich Entwürfe für Informatikunterricht im Kontext ebenfalls an den Grundsätzen und Standards, d.h. in jedem Entwurf wird festgehalten, welche Kompetenzen gefördert werden. Langfristig wird angestrebt, jede Kompetenz der Bildungsstandards abzudecken, um somit kontextorientierte Lehrmaterialien für einen umfassenden Informatikunterricht zur Verfügung zu stellen. Zu jedem Unterrichtsentwurf gehört demnach ein Verweis auf die Kompetenzen der Inhalts- und Prozessbereiche, die mit den entsprechenden Kontexten gestärkt werden. Beispiele für kontextorientierte Unterrichtsentwürfe finden sich auf der IniK-Website [KO09].

Die Fachprinzipien werden über Dekontextualisierung frei gelegt. Der in den Bildungsstandards formulierte kompetente Umgang mit diesen Prinzipien kann anschließend rekontextualisiert, d.h. in anderen Kontexten überprüft werden.

Das pädagogische Gegenstromprinzip der Orientierung sowohl an Ereignissen aus der Lebenswelt (bottom-up) als auch an Fachprinzipien und Kompetenzen (top-down) zeichnet den kontextorientierten Unterricht aus.

2.3 Methodische Vielfalt

Methodische Vielfalt richtet sich gegen einen Methodenzwang in pädagogischen Situationen, der in dieser Form natürlich nie bestanden hat, sondern mit seinem Schwerpunkt des Frontalunterrichts vielmehr Ausdruck von Zeitnot bei der Vorbereitung und Bequemlichkeit bei der Durchführung ist. Die Umgestaltung des Unterrichts zu einem stärkeren lebensweltlichen Bezug sollte aber nicht nur inhaltlich erfolgen, sondern die Möglichkeit der stärkeren Partizipation der Schüler berücksichtigen. Gerade die Informatik bietet mit ihrer Aufgeschlossenheit gegenüber Unterrichtsmedien ein breites Experimentierfeld für verschiedene Unterrichtsmethoden.

Neben dem Klassiker von Hilbert Meyer [ME87] sei an dieser Stelle auf das breite Angebot an Methodenbeschreibungen in [HU09] hingewiesen, das zumindest erste Anregungen vermittelt, was alles möglich ist. Auch kontextorientierte Entwürfe wie die Unterrichtsreihe zu Chatbots [WH09] zeigen, wie sich die Forderung nach Methodenvielfalt konkretisieren lässt.

Beispiele für geeignete Unterrichtsmethoden sind:

- Kleine Softwareprojekte, die von den Besonderheiten des Kontexts ausgehen.
- Experimente und Erkundungen des Kontexts.
- Das Suchen, Finden und diskutieren geschichtlicher Zusammenhänge, um den Einfluss von Informatiksystemen auf den Kontext zu verdeutlichen.
- Recherche, ggf. in arbeitsteiligen Gruppen.

3 Unterrichtsplanung

Die Autoren des Lehrerleitfadens von CHiK schlagen zwei Wege vor, kontextorientierten Unterricht zu planen [IP05]:

1. Ausgehend von Fachkonzepten und Standards werden geeignete Kontexte gesucht, deren Bearbeitung die Unterrichtsziele ermöglichen. Zu diesen Kontexten werden Hintergrundinformationen und weiteres Material gesammelt, auf dessen Grundlage anschließend der Unterricht geplant wird.
2. Ein interessanter Kontext wird zunächst inhaltlich recherchiert und erschlossen. Anschließend werden die in ihm enthaltenen Fachprinzipien und Basiskonzepte zusammengestellt, die zu diesem Kontext gehören. Zwar sollten sich mit jedem Kontext alle Basiskonzepte verbinden lassen, notwendigerweise stehen aber im

Unterricht einige stärker im Vordergrund als andere. Mit diesen Zielen wird anschließend die Unterrichtseinheit geplant.

Darüber hinaus kann auf eine wachsende Sammlung erprobter Unterrichtsentwürfe zurückgegriffen werden, die auf der Website www.informatik-im-kontext.de zusammengetragen werden [KO09].

Kontextorientierte Unterrichtsentwürfe

Entwürfe für kontextorientierte Unterrichtseinheiten haben folgende Struktur:

- Analyse des Kontexts unter Berücksichtigung der Bedeutung für die Gesellschaft sowie für die Lebenswelt der Schülerinnen und Schüler.
- Kompetenzerwartungen: Bewährt hat sich hier eine Liste derjenigen Inhalts- und Prozessbereiche, die durch die Unterrichtseinheit gestärkt werden.
- Mögliche Dekontextualisierungen: Je nach Auswahl der lebensweltlichen Dimensionen, die im Unterricht thematisiert werden, können verschiedene Prinzipien herausgearbeitet werden.
- Planen der Unterrichtsphasen: [IP05] schlägt vier Phasen vor:
 - Begegnungsphase
 - Neugier- und Planungsphase
 - Erarbeitungsphase
 - Vernetzungs- und Vertiefungsphase

Das Schema hat sich bei verschiedenen IniK-Unterrichtsentwürfen als hilfreich, bei anderen als unbrauchbar erwiesen. Insofern ist es als Anregung aufzufassen und sollte nicht den normativen Charakter einer *Stundenfigur* erhalten.

- Verknüpfungen: Kontextorientierter Unterricht ist dem Wesen nach fachübergreifend und fächerverbindend. Auch wenn dieser Unterrichtskombination die bereits erwähnten und an anderer Stelle ausführlicher diskutierten Hindernisse im Wege stehen, sollten zumindest mögliche Verbindungen aufgezeigt und skizziert werden.

4 Chatbots als Beispiel für eine kontextorientierte Unterrichtseinheit

Die kontextorientierte Unterrichtseinheit „Chatbots“ ist ausführlich dokumentiert ([WH09], Online bei [KO09]). Im Folgenden soll daher nur kurz dargestellt werden, wie die oben dargelegten Prinzipien in dieser Beispieleinheit umgesetzt werden.

Wir haben weiter oben „Kontext als Menge von lebensweltlichen Themen bzw. Fragestellungen“, definiert, die von den Schülerinnen und Schüler als zusammenhängend und sinnstiftend erfahren werden. Im Fall der Chatbots geht es ganz allgemein um Computer (bzw. Roboter), mit denen man sich „unterhalten“ kann. Diese Situation ist den Lernenden aus der Science-Fiction wohl bekannt, das Chatten ist unter Jugendlichen sehr beliebt. Sie haben auch die Erfahrung gemacht, dass man heutzutage bei einer Hotline gezwungen wird, mit einem in der Regel wenig „intelligenten“ Computer zu reden und dass es häufig sehr schwierig ist, am Computer vorbei einen menschlichen Mitarbeiter zu erreichen.

Weniger vertraut ist der Gedanke, dass auch der Chatpartner ein Roboter sein könnte. Ebenso wenig ist der aktuelle Entwicklungsstand und die technische Funktionsweise realer Chatbot-Systeme bekannt. Auch die Vorläufer der heutigen Chatbots – theoretisch mit Turings Test (1950) vorweggenommen, praktisch erstmals mit Weizenbaums Eliza (1966) umgesetzt – sind den Schülerinnen und Schülern weithin unbekannt.

In der Chatbots-Reihe konfrontieren die Schülerinnen und Schüler ihre Vorstellungen mit realen Informatiksystemen. Die Fähigkeit, sich in natürlicher Sprache verständlich zu äußern, wurde schon von Turing als so zentral erkannt, dass er sie zur Grundlage seiner Definition von Maschinenintelligenz gemacht hat. Nach den vorliegenden Erfahrungen motivieren diese Fragen die Lernenden über den Verlauf der gesamten Unterrichtsreihe sehr stark.

Ausgehend von der Mensch-Maschine-Kommunikation wurden Dekontextualisierungen in folgende Dimensionen vorgenommen: technisch (wie funktioniert ein aktueller Chatbot?), ethisch (sollten Chatbots eingesetzt werden?), ökonomisch (warum werden Chatbots eingesetzt?) und historisch (welche Rolle spielte die Idee der Chatbots in der Geschichte der KI?).

Aus der detaillierten Beschreibung der Unterrichtsreihe [WH09] ergibt sich, dass alle fünf Inhaltsbereiche abgedeckt werden, Schwerpunkte bilden die Bereiche I4 und I5. Auch aus allen Prozessbereichen finden sich Standards, deren Erwerb in der Chatbot-Reihe befördert wird; Schwerpunkte bilden hier die Bereiche P2 und P4 (s. Abschnitt 2.2).

Bei der Entwicklung von IniK-Unterrichtseinheiten kann es aber nicht nur darum gehen, möglichst viele der Inhalts- und Prozessbereiche aus den Bildungsstandards Informatik abzudecken. Konkreter Unterricht muss auch daraufhin geprüft werden, ob die zu erarbeitenden Kompetenzen durch die Standards erfasst werden oder ob evtl. weitere wichtige Kompetenzen in den Standards fehlen oder andere entfallen können, weil sie keine Mindeststandards darstellen. In diesem Sinn dient IniK auch als „proof of concept“ der Informatik-Standards, die es weiterzuentwickeln gilt.

In der Chatbots-Reihe ist ein Schwerpunkt die Geschichte der KI. Leider finden sich in den GI-Standards keinerlei Kompetenzen, die sich auf die Geschichte der Datenverarbeitung beziehen. Um diese und andere Leerstellen zu füllen, haben sich die Autoren der Unterrichtseinheit beim aktuellen Berliner Rahmenlehrplan ITG/Informatik (Sek I) be-

dient. Im Einzelnen wurden die folgenden Kompetenzen in der Auflistung der Standards vermisst (Quellennachweise in [WH09]):

Die Schülerinnen und Schüler

- *verfügen über grundlegende Kenntnisse zur historischen Entwicklung der Informatik*
- *erfahren die extreme Begrenztheit der „Verständnisfähigkeit“ der Maschine beim Entwurf und Test eigener Problemlösungen*
- *kommunizieren in Rollenspielen; sie nehmen dabei ihrer Rolle entsprechende Standpunkte ein*

Als Unterrichtsmethoden und Sozialformen kamen neben den „Klassikern“ Partnerarbeit am Rechner, Gruppenarbeit und Unterrichtsgespräch u. a. Diskussionen im Plenum, ein Mini-Projekt, ein Gruppenpuzzle sowie ein Rollenspiel zum Einsatz.

5 Fazit

Informatik im Kontext greift die bekannten Forderungen nach der Gestaltung von Unterricht auf, die sich sowohl an lebensweltlichen Kontexten als auch an den überzeitlichen Standards der Fachdisziplin orientieren soll. Die Integration dieser zwei gegenläufigen Ziele kann auf verschiedene Weise erfolgen. Informatik im Kontext wählt eine systematische Ausrichtung an Kontexten, die im Zentrum des Unterrichts steht und um die sich die übrigen didaktischen Entscheidungen anordnen.

Die IniK-Prinzipien Orientierung an Kontexten und Dekontextualisierung des Unterrichtsgegenstands, Orientierung an Standards sowie Methodenvielfalt bilden zusammen mit der vorgeschlagenen Struktur für kontextorientierte Unterrichtsentwürfe eine Reihe von Hilfsmitteln, um diese systematische Ausrichtung zu erreichen.

Literaturverzeichnis

- [A78] Arlt, Wolfgang (Hrsg.): EDV-Einsatz in Schule und Ausbildung. München: Oldenbourg, 1978.
- [BA07] Bayrhuber, Horst et.al: Biologie im Kontext MNU 60/5, S. 282-286.
- [EL06] Elster, Doris: Kontexte als Strukturelement des Unterrichts. bik-Handreichung für die Praxis. Online (2009): http://bik.ipn.uni-kiel.de/typo3/fileadmin/user_upload/pdf/HR2_26.06.06.pdf
- [EN04] Engbring, Dieter: Informatik im Herstellungs- und Nutzungskontext. Paderborn (Dissertation) 2004. Online (2009): <http://deposit.ddb.de/cgi-bin/dokserv?idn=975810227>
- [GI08] GI e.V. (Hg.): Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. Beilage zu LOG IN, 28. Jg. (2008), Heft Nr. 150/151.

- [FSR07] di Fuccia, David, Schellenbach-Zell, Judith; Ralle, Bernd: Chemie im Kontext. Entwicklung, Implementation und Transfer einer innovativen Unterrichtskonzeption. In: MNU 60/5, S. 274-282.
- [GR76] Gruppe Informatik in Berliner Schulen (Beer/Koerber/Reker/Sack/Schulz): Stand der Curriculum-Entwicklung für den Informatik-Unterricht in Berlin. Berlin, 1976.
- [HU09] Hupfeld, Walter (Hg.): Methodensammlung - Anregungen und Beispiele für Moderatoren. Online (2009): <http://www.learn-line.nrw.de/angebote/methodensammlung/index.php>
- [IP05] Leibniz-Institut für die Pädagogik der Naturwissenschaften (IPN) (Hg.): Chemie im Kontext. Hinweise zur Konzeption, Durchführung und Evaluation von Unterrichtseinheiten. Online (2009): <http://www.chik.de/dateien/lehrerleitfaden.pdf>
- [IP07] Leibniz-Institut für die Pädagogik der Naturwissenschaften (IPN) (Hg.): Physik im Kontext (piko). Online (2009): <http://www.uni-kiel.de/piko/index.php?topic=9>
- [KK07] Koubek, Jochen; Kurz, Constanze: Informatik-Mensch-Gesellschaft im Schulunterricht. In: Schubert (Hg.): Informatische Bildung in Theorie und Praxis. Siegen, 2007. S. 125-134.
- [KO05] Koubek, Jochen: Informatische Allgemeinbildung. In: Friedrich (Hg.): Unterrichtskonzepte für informatische Bildung. Dresden, 2005. S. 57-66.
- [KO09] Koubek, Jochen (Hg.): Informatik im Kontext. Online (2009): informatik-im-kontext.de
- [ME87] Meyer, Hilbert: Unterrichtsmethoden, 2 Bd. Cornelsen, 1987.
- [MU95] Muckenfuß, Heinz: Lernen im sinnstiftenden Kontext. Entwurf einer zeitgemäßen Didaktik des Physikunterrichts. 358 S., Berlin: Cornelsen 1995.
- [MSD07] Mikelskis-Seifert, Silke; Duit, Reinders: Physik im Kontext – Innovative Unterrichtsansätze für den Schulalltag. MNU 60/5, S. 265-274.
- [SZ78] Schulz-Zander, Renate: Analyse curricularer Ansätze für das Schulfach Informatik (in [A78], S. 40-49.
- [WE01] Weinert, F. E. (2001). Vergleichende Leistungsmessung in Schulen – eine umstrittene Selbstverständlichkeit. In: F. E. Weinert (Hg.): Leistungsmessungen in Schulen. Weinheim/ Basel, S. 17-31.
- [WH09] Witten, Helmut; Hornung, Malte: Chatbots. Eine Unterrichtsreihe zu „Informatik im Kontext“ (IniK), Teil 1. LOG IN Heft Nr. 154/155 (2008).
- [WI53] Wittgenstein, Ludwig: Philosophische Untersuchungen, 1953.

Rote Fäden und Kontextorientierung im Informatik-Unterricht

Arno Pasternak

Technische Universität Dortmund, Fakultät für Informatik, und
Fritz-Steinhoff-Gesamtschule Hagen
arno.pasternak@cs.tu-dortmund.de

Jan Vahrenhold

Technische Universität Dortmund, Fakultät für Informatik
jan.vahrenhold@cs.tu-dortmund.de

Abstract: Die *Bildungsstandards Informatik für die Sekundarstufe I* formulieren Ansprüche an den Informatik-Unterricht für die Jahrgänge 5 bis 10. Ausgehend von diesen Standards erscheint eine Unterrichtskonzeption einer *Informatik im Kontext* naheliegend und viel versprechend. Sie ist aber nur dann zielführend, wenn nicht nur einzelne, sondern wesentliche Teile des Curriculums in dieser Form organisiert sind und dabei im Gesamtzusammenhang für den Schüler deutlich wird, welche informatischen Konzepte vermittelt werden.

Dieser Beitrag schlägt vor, ein *Konzept der roten Fäden* für den Informatik-Unterricht in der Sekundarstufe I zu entwickeln und zu realisieren. Diese roten Fäden sollen für Schüler wie Lehrer die Orientierung im gesamten Informatik-Unterricht der Sekundarstufe I darstellen und ggf. einen bruchfreien Übergang in den Informatik-Unterricht der Sekundarstufe II ermöglichen.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): *Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«*. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21. - 24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Programming-Wiki: Online programmieren und kommentieren

Michael Hielscher
Zentrum für Bildungsinformatik
Pädagogische Hochschule Bern
mail@michael-hielscher.de

Christian Wagenknecht
Fachbereich Informatik
Hochschule Zittau/Görlitz
c.wagenknecht@hs-zigr.de

Abstract: In Zeiten von Web 2.0 finden immer mehr Aktivitäten orts- und zeitunabhängig über das Internet statt. Lerninhalte und Aufgabentexte in einem Wiki aufzubereiten, um sowohl Schülern und Schülerinnen als auch anderen Lehrpersonen die gemeinsame Nutzung zu ermöglichen, ist keine Seltenheit mehr. Hier wird beschrieben, wie Wikis durch Software-Erweiterungen auch als interaktive Lernumgebungen genutzt werden können. Dies wird am konkreten Beispiel „Einführung in die Programmierung“ gezeigt.

1 Einleitung

Wikis werden in erster Linie als Werkzeug für die computerunterstützte Mensch-Mensch-Kommunikation verwendet, um verschiedenste Materialien in einer Nutzergemeinde kollaborativ zu erarbeiten und zu verteilen. Unter dem Begriff Wiki verstehen wir hier die Verknüpfung aus einem technischen Wikisystem (Wiki-Engine) und den darin eingestellten Inhalten. Ein sehr populäres Wiki ist die Wikipedia. Sie basiert auf dem MediaWiki-System (vgl. [Med08]).

Neben dem stetig wachsenden Einsatz in der Industrie, sind Wikis auch in Hochschulen und Schulen gewinnbringend einsetzbar (vgl. [Döb07]). Lehrinhalte lassen sich in Wikis aufbereiten und je nach Wunsch des Autors gemeinsam mit anderen Lehrpersonen entwickeln. Ein gutes Beispiel ist das ZUM-Wiki (vgl. [ZUM08]) mit mehr als 1500 angemeldeten Benutzern, welches Informationen und Unterrichtsmaterialien zu praktisch allen Fächern der Sekundarstufe 1 und 2 zusammenträgt – eine Plattform von Lehrpersonen für Lehrpersonen.

Lerninhalte, die auf Wikis für den Unterricht bereitgestellt werden, lassen sich von Schülerinnen und Schülern nicht nur in der Schule, sondern auch vom heimischen Computer aus bearbeiten, kommentieren und ggf. modifizieren. Die Bedienanforderungen an Wiki-Autoren sind sehr gering, sodass auch Lernende Inhalte mit wenig Aufwand publizieren können. Das für traditionelle Webseiten typische Hochladen von Dateien auf den Webserver entfällt. Eine ausführliche Zusammenschau der Vorteile der Wiki-Technologie ist beispielsweise in [KH08] enthalten. Eine Aufgabenstellung wie „Verfasse eine Wikiseite zum Bubblesort-Algorithmus und präsentiere sie anschließend deinen Mitschülern.“ kann unter dem Aspekt *Lernen durch Lehren* vielversprechend sein.

Im Vergleich zum kooperativen Ansatz eines Wikis sind Learning Management Systems (LMS) eher zentralistisch organisiert. Sie beruhen auf der Vorstellung, dass Lernende vorbereitetes Lehrmaterial konsumieren, ohne es modifizieren (ergänzen, verändern, diskutieren) zu können. Ein LMS verfügt im Allg. über eine „eingebaute“ starre Didaktik, die nur mit größerer Mühe an die Wünsche guter Autoren angepasst werden kann. LMS sprechen daher eher Content-Provider statt Lehrende mit fach- und mediendidaktischer Qualifikation an. Um diesen didaktischen Defiziten entgegen zu wirken, werden Wikis in LMS integriert. Im Gegensatz dazu bevorzugen wir die Verwendung eines speziell präparierten Wikis als LMS-freies Basissystem mit unstrukturierter didaktischer Gestaltungsmöglichkeit für qualifizierte Lehrpersonen.

Neben der Mensch-Mensch-Kommunikation lässt sich der Computer auch für die Mensch-Maschine-Kommunikation nutzen. Wir sprechen von Lernsoftware, bei der Schülerinnen und Schüler in der Interaktion mit dem Computer etwas lernen sollen. Bei dieser Nutzung ist der Grad der Interaktivität zwischen dem Lernenden und der Lernumgebung entscheidend. Höhere Interaktion ist aufwändig in der Produktion und oft nur für gut formalisierbare Themen umsetzbar (vgl. [RH04]). Klassische LMS bieten für diesen Zweck interaktive Elemente, die jedoch nur tiefe kognitive Interaktionsstufen (vgl. [Sch02]) adressieren (Multiple-Choice-Fragen, Lückentexte usw.).

Zur Anreicherung eines Wikis mit interaktiven Elementen haben wir entsprechende Software-Erweiterungen in eine konventionelle Wiki-Engine eingebaut. Im Folg. zeigen wir am Beispiel der Einführung ins Programmieren, wie derart *angereicherte Wikis* als interaktives Lernmedium eingesetzt werden können. Dieser Ansatz lässt sich auch auf andere Themengebiete, etwa aus der Mathematik, Physik, Chemie oder Wirtschaft, übertragen.

Ursprünglich primär als kollaborative Arbeitsplattform gedacht, können sich Wikisysteme mit diesen Erweiterungen noch mehr in Richtung *Autorenwerkzeug für interaktive Lernumgebungen* entwickeln.

2 Programming-Wiki

Zentrale Inhalte des Programmierunterrichts sind das Lesen und Schreiben von Code sowie das Testen entwickelter Programmen. Nicht in jedem Fall wird ein Programm auf einem leeren Blatt entworfen. Lehrpersonen können Programmgerüste oder geeignete Bausteine vorgeben, um den Entwurfsprozess anzuleiten. Die für den Lehrenden durchaus mühevoll verteilte solchen Materials, sei es auf Papier (Arbeitsblatt) oder in digitaler Form (Webseite, LMS, etc.), kann eine echte Unterstützung des Lernenden nicht garantieren: Oftmals ist der Zusammenhang zwischen Aufgabenstellung, Arbeitsanleitung, Lehrtext und eben diesen Bausteinen für den Lernenden nicht transparent oder geht im Bearbeitungsprozess unter. Außerdem muss Code in die verwendete Entwicklungsumgebung übertragen werden, was einen Medienbruch bedeutet.

Die Installation und Konfiguration einer Programmierumgebung auf dem eigenen Computer ist für Schülerinnen und Schüler aufwändig und erfordert häufig Fachwissen. Stimmen die lokalen Einstellungen nicht mit denen in der Schule überein, ergeben sich bei der Ausführung von Programmen, die im Unterricht erarbeitet wurden, schnell schwer zu

interpretierende Fehlermeldungen.

Im Trend von Web 2.0 entstehen immer mehr Online-Entwicklungsumgebungen (z.B.: [Her08]). Der Entwurf solcher Umgebungen steht aber erst am Anfang und sie sind bislang nicht für den Schuleinsatz konzipiert. Man darf nicht übersehen, dass komplexere Arbeitsumgebungen einen nicht zu unterschätzenden Einarbeitungsaufwand für die Nutzer erfordern. Gerade in der Anfangsphase des Programmierunterrichts stehen kurze Programme im Zentrum. Dafür würde eine einfache Entwicklungsumgebung ausreichen. Die Nutzung eines angereicherten Wikis als *Programming-Wiki* bietet sich an.

Mittels einer Software-Erweiterung (Extension) wird das bekannte MediaWiki-System zur Integration von interaktiven Elementen befähigt. Dadurch kann ein eingebetteter Programmtext im Wiki nicht nur gelesen, sondern aktiv verändert, gespeichert und ausgeführt werden. Wir bezeichnen diese Wikiseiten deshalb als *interaktive Arbeitsblätter*. Ein Interpreter der gewählten Programmiersprachen (derzeit Java, Pascal, Scheme, JavaScript, XSLT und XPath) arbeitet im Hintergrund, um Programmtexte zu evaluieren und die Ergebnisse der Abarbeitung wieder im Arbeitsblatt zu präsentieren. All dies geschieht unmittelbar und innerhalb des Webbrowsers mit Hilfe kleiner Java-Applets, die die nötigen Interpreter ohne Installation auf dem Clientrechner realisieren. Voraussetzung ist ausschließlich ein Java-fähiger Webbrowser. Derzeit werden Firefox und Internet Explorer unterstützt. Es braucht keine weitere Software installiert zu werden.

Aufgabe: Glücksspiel

[Bearbeiten]

Bei einem Glücksspiel geht es darum, mit 3 Würfeln möglichst viele gleiche Augen zu werfen. Entwickle ein Programm, welches dreimal würfelt und die Augenzahl ausgibt (zum Beispiel 6,3,4). Anschließend soll ausgegeben werden, ob der Spieler verloren oder gewonnen hat.



Gewinnbedingung: Mindestens 2 Würfel haben die gleiche Augenzahl.

```
1 int w1,w2,w3;
2 w1 = (int) (Math.random()*6)+1;
3 w2 = (int) (Math.random()*6)+1;
4 w3 = (int) (Math.random()*6)+1;
5 System.out.println("Du hast " +w1+", "+w2+", "+w3+" gewürfelt.");
6 if (w1 == w2 )
7   System.out.println("Du hast gewonnen!"); else
8   if (w1 == w3 )
9     System.out.println("Du hast gewonnen!"); else
10    if (w2 == w3 )
11      System.out.println("Du hast gewonnen!"); else
12      System.out.println("Du hast verloren!");
13
```

speichern & ausführen

```
> Du hast 3,6,3 gewürfelt.
  Du hast gewonnen!
```

Abbildung 1: Beispiel einer Programmieraufgabe (Schülerlösung)

Die Idee des Programming-Wiki ist im Kern nicht neu: Bereits 1984 beschrieb Donald Knuth seinen *Literate-programming*-Ansatz (vgl. [Knu84]), bei dem es darum geht, sowohl die Dokumentation als auch den Quelltext eines Programms in einer gemeinsamen Datei abzulegen. In unserem Fall geht es analog um die Verknüpfung von Lehrtexten und Quelltexten innerhalb einer Webseite. Seit mehreren Jahren wird dieser Ansatz an der Hochschule Zittau/Görlitz im Kurs Programmierparadigmen praktiziert. 2002 wurde zu diesem Zweck SchemeNet (vgl. [Wag05], [WB05]) entwickelt, welches es erlaubt, Programmtexte der Programmiersprache Scheme innerhalb einer Webseite einzubinden, zu editieren und direkt auszuführen. SchemeNet-Seiten werden vom Lehrenden in einer eigenen XML Sprache verfasst und in HTML transformiert. Für die Ausführung von Scheme-Programmen innerhalb einer solchen Webseite wird ein „Scheme-Server“ verwendet, der für jeden Nutzer separat die Rolle des Scheme-Interpreters übernahm. 2007 wurde der Scheme-Server durch einen Client-seitigen Scheme-Interpreter ersetzt, um Sicherheits- und Performanceprobleme zu lösen. Das System wurde über mehrere Jahre erfolgreich mit großem Interesse der Studierenden erprobt und eingesetzt. Die Herstellung von derartigen interaktiven Arbeitsblättern blieb einer kleinen Autorentgemeinde mit der Bereitschaft, eine speziellen XML Sprache zu erlernen, vorbehalten. Die generierten HTML-Dateien mussten manuell auf einen Webserver hochgeladen werden, um sie den Lernenden zur Verfügung zu stellen. Das in diesem Beitrag vorgestellte Programming-Wiki stellt eine Weiterentwicklung des beschriebenen Ansatzes dar. Durch die konsequente Nutzung der Wiki-Technologie wird, neben vielen weiteren Vorteilen, die Handhabung für Autoren maßgeblich verbessert.

3 Programming-Wiki aus der Sicht des Lernenden

Zur Illustration der Lernenden-Perspektive auf ein Programming-Wiki verwenden wir in den folgenden Beispielen und Abbildungen jeweils nur Auszüge aus interaktiven Arbeitsblättern. Es wird speziell auf die interaktiven Elemente innerhalb umfangreicher Lehrtexte eingegangen. Dabei wurden bewusst klassische und kurze Beispiele aus der Mathematik und Geometrie gewählt, um den Blick auf die Realisierung im Programming-Wiki zu lenken. Im Anhang finden sich kreative Beispiele zur Ton- und Grafikerzeugung. Als Sprache wurde exemplarisch Java gewählt.

In Abb. 1 ist eine Übungsaufgabe zur Erzeugung von Zufallszahlen und Textausgabe dargestellt. Die Schülerin hat bereits eine Lösung im Eingabebereich, der *Codebox*, erarbeitet und diese über die Schaltfläche abgespeichert und ausgeführt. Das Ergebnis wird im nachfolgenden Ausgabefeld präsentiert.

In vielen Fällen werden vom Lehrer bereits Programmteile vorgegeben, die vom Lernenden ergänzt werden sollen. Dies können etwa Methodenrumpfe als Schnittstellenbeschreibungen, Beispielaufrufe oder vollständige Programme zum Experimentieren sein.

Abb. 2 zeigt eine Aufgabe zur Entwicklung des Bubble-Sort-Algorithmus. Die von der Lehrperson gelieferte Vorgabe kann bereits vom Schüler ausgeführt und die Ausgabe betrachtet werden. Da die Methode `BubbleSort` keine Anweisung enthält, wird die vordefinierte Liste unverändert ausgegeben.

Aufgabe: Bubble-Sort

[Bearbeiten]

In dieser Aufgabe sollst du den BubbleSort Algorithmus implementieren. Im nachfolgenden Eingabefeld ist bereits eine passende Methode für dich vorgegeben. Eine Liste aus Ganzzahlen wird als Parameter übergeben. Die Methode besitzt keinen Rückgabewert, da die Liste als Referenz übergeben und verändert wird.

```
1 void BubbleSort (int[] liste) {  
2     // Implementiere den BubbleSort Algorithmus  
3 }  
4
```

```
1 int[] testliste = new int[]{1,2,9,5,2,23,1};  
2 BubbleSort (testliste);  
3  
4 for (int i=0; i < testliste.length; i++)  
5     System.out.print(testliste[i]+",");  
6  
7
```

speichern & ausführen

> 1,2,9,5,2,23,1,

Abbildung 2: Vorgabe für eine Aufgabe zu Bubble-Sort

Je nach Aufgabenart können Arbeitsblätter auch *Feedback-Elemente* enthalten. Vom Lehrer definierte Testfälle werden automatisch überprüft. Abb. 3 illustriert dies am Beispiel des Bubble-Sort-Algorithmus'. Es geht nicht um eine echte Lösungskontrolle, sondern um eine einfache, qualitative Erfolgsbeurteilung für die Lernenden.

Die ggf. modifizierten Programmtexte in den Codeboxen und die Ergebnisse von Feedback-Elementen werden für jeden angemeldeten Benutzer separat in der Wiki-Datenbank gespeichert. Wechselt eine Schülerin ihren Arbeitsplatz im Computerpool, oder möchte von zu Hause ihre Lösung komplettieren, findet sie im Wiki stets ihren aktuellen Arbeitsstand vor. Ein Datentransfer von einem Rechner zum nächsten entfällt. Da keine lokalen Daten auf dem Arbeitsrechner abgelegt werden, können diese auch nicht verloren gehen, etwa durch Speichern auf einem falschem Datenträger oder unter einem falschem Dateinamen.

Die Navigation innerhalb von Lehrtexten und Aufgabenblättern wird durch das Wiki bereitgestellt. Eine Volltextsuche wird ebenso angeboten wie eine Versionsgeschichte und eine ausgereifte Benutzerverwaltung, über die sich jeder Schüler zu Kursbeginn anmelden kann.

4 Programming-Wiki aus der Sicht des Lehrenden

Das Konzept und die Handhabung zum Anlegen und Editieren von Wiki-Seiten wird unverändert aus dem MediaWiki-System übernommen und ist den meisten Lehrenden durch die Wikipedia vertraut. Der Autor von Wiki-Inhalten kann neben Text, Tabellen, Grafiken, Audiodateien (MP3), Videos, ZIP-Archiven oder PDF-Dokumenten, beliebig viele Codeboxen und Feedback-Elemente in eine Wikiseite einfügen. In einer frei editierbaren Code-



Abbildung 3: Schülerlösung der Übungsaufgabe zu Bubble-Sort

box können fertige Quelltexte zum Experimentieren, Teillösungen zur Vervollständigung oder ausschließlich Hinweise zur Entwicklung einer eigenen Lösung als Vorgabe hinterlegt werden. Für ein Feedback-Element wird hingegen ein Testausdruck angegeben, welcher auf die Lösung des Schülers bei Betätigen der Prüfschaltfläche angewendet wird.

Um ein interaktives Element einzufügen, wird in den Quelltext des Arbeitsblatts ein XML-Tag wie `<eval>...</eval>` eingefügt. Das Wiki erstellt für den Betrachter der Seite eine geeignete Darstellung etwa in Form eines Editors inklusive Syntax Highlighting (vgl. Abb. 4).

Das Programming-Wiki unterstützt neben der reinen Textausgabe auch graphische und akustische Ausgaben. In Abb. 4 wird ein Turtle für das Zeichnen eines Kreises eingesetzt. Als Ausgabe wird ein *Canvas-Element* verwendet, welches vom Autor mit dem Tag `<canvas>` unterhalb eines `<eval>` Elements platziert wurde. In diesem Beispiel wird eine Lösung bereits vorgegeben, mit der die Schüler und Schülerinnen experimentieren können.

Ein Feedback-Element kann über das Tag `<check>` eingebunden werden. In Abb. 5 wird dies am Beispiel der Fibonacci-Funktion gezeigt. Die Methode `fibonacci` wird hierfür mit mehreren Eingabewerten aufgerufen und mit Referenzwerten verglichen.

Da alle Schülerlösungen in der Wiki-Datenbank abgelegt werden, können diese vom Lehrenden ausgewertet werden. Meldet sich eine Lehrperson am System an, werden alle Benutzernamen aufgelistet, die bereits Lösungen für das aktuelle Arbeitsblatt erarbeitet haben. Verwendet der Schüler die enthaltenen Feedback-Elementen zur Beurteilung seiner Lösungen, so wird angezeigt, ob er erfolgreich war oder nicht (siehe Abb. 6). Durch einen Klick auf einen Namen werden die Lösungen dieses Nutzers geladen und können vom Lehrenden begutachtet und ggf. auch kommentiert oder korrigiert werden. Betrachtet der



Abbildung 4: Eine Codebox in ein Arbeitsblatt einbinden



Abbildung 5: Hinzufügen eines Feedback-Elements

Schüler anschließend erneut sein Arbeitsblatt, kann er die Anmerkungen des Lehrenden einsehen und entsprechend darauf reagieren.



Abbildung 6: Benutzerauswahl für den Administrator

5 Zusammenfassung und Ausblick

Am konkreten Beispiel der Vermittlung von Programmiergrundlagen haben wir gezeigt, wie interaktive Elemente in ein Wiki-System integriert werden können. Damit lassen sich die Vorteile eines Wikis mit denen von spezialisierten Lernumgebungen verknüpfen und Synergieeffekte nutzen. Der Erfolg von Wikis ist nicht zuletzt ihrer einfachen und leicht zu erlernenden Bedienung zuzuschreiben. Wikis, wie das verwendete MediaWiki, haben eine breite Akzeptanz und sind kostenfrei erhältlich.

Das vorgestellte Konzept lässt sich auch auf andere Fachbereiche übertragen. In weiteren Projekten sollen interaktive Elemente zur Mathematik, Chemie aber auch zum Fremdsprachenunterricht entwickelt werden. Unser Interesse gilt hier vor allem Inhalten, die gut formalisierbar sind.

Unter der Adresse: <http://www.michael-hielscher.de/wiki> steht das hier beschriebene Programming-Wiki allen Interessenten zum Experimentieren offen. An der Implementierung der Programming-Wiki-Erweiterungen waren die Studierenden Anna Prenzel, Philipp Herzig, Filip Martinovský dankenswerterweise beteiligt.

Anhang: Weitere Beispiele aus Programming-Wiki

Im Folg. illustrieren wir einige ausgewählte Programming-Wiki-Beispiele, die im Programmierunterricht von Schülerinnen und Schülern vollständig oder teilweise bearbeitet wurden. In den entsprechenden Codeboxen (s. Screenshots) sind jeweils zugehörige Lösungen angegeben.

Abb. 7 zeigt ein Beispiel zu John Conway's Game of Life, bei dem Arrays für das Spielfeld zum Einsatz kommen. Die Abarbeitung des von Conway angegebene Algorithmus' führt zu einer graphischen Darstellung, die im Programming-Wiki nahtlos aufgenommen wird.

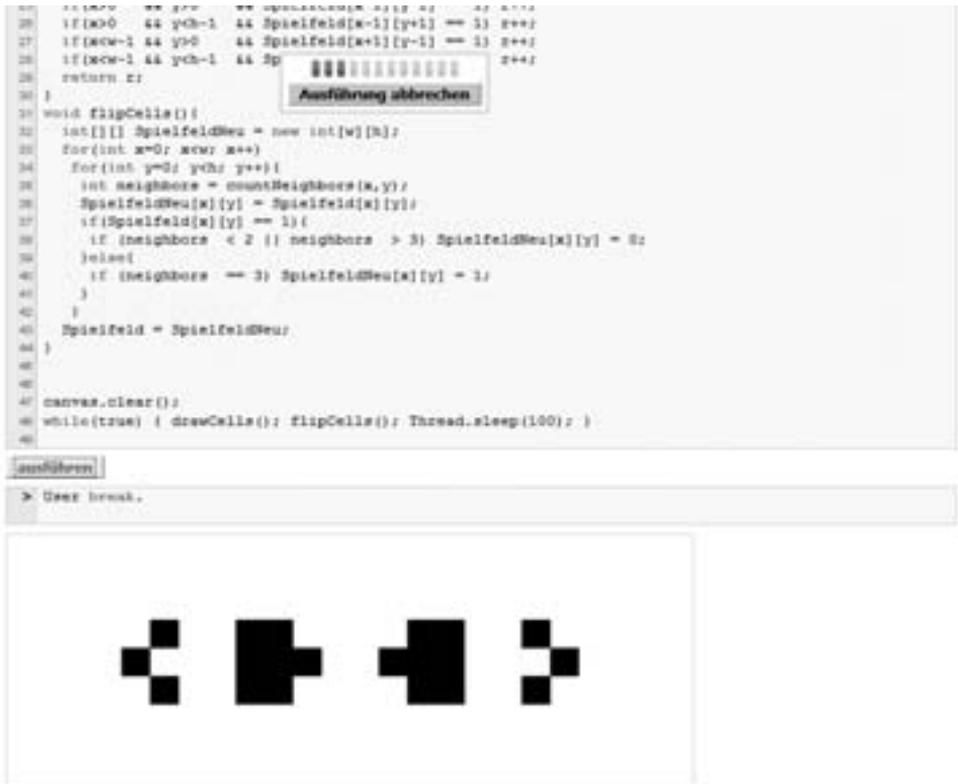


Abbildung 7: Beispiel zu John Conway's Game of Life

Abb. 8 enthält ein Einstiegsbeispiel zur imperativen Programmierung. Die Schülerinnen und Schüler experimentieren mit vordefinierten Prozeduren (Befehlen), um eine Spielfigur durch ein Labyrinth zu navigieren. Befehlsfolgen können zusammengefasst und unter einem neuen Namen gespeichert werden. Auf diese Weise konstruieren die Lernenden neue Prozeduren, wie etwa in folgendem Beispiel:

```
void laufe3nachoben() {
    nachoben();
    nachoben();
    nachoben();
}
```

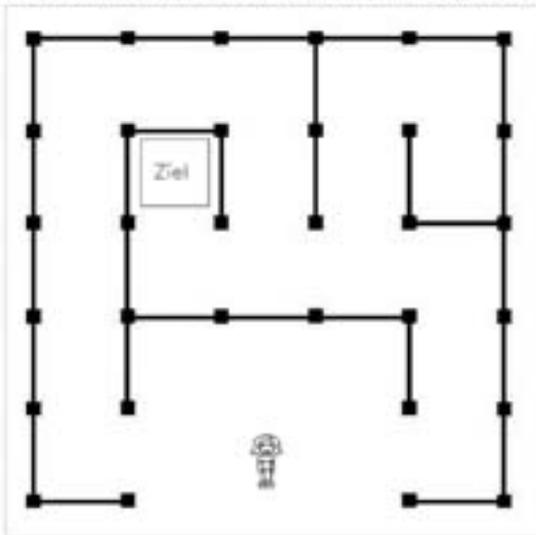
Der Weg aus dem Labyrinth

[Bearbeiten]

Anna ist in einem Labyrinth gefangen und du musst ihr helfen. Anna kann über 4 **Methoden** gesteuert werden. Wende diese nacheinander an, um Anna bis zum rot markierten Ziel zu lenken.

Die Methode `start()` stellt Anna wieder an den Anfang des Labyrinths. Wir rufen diese Methode deshalb zu Beginn von jedem Programmstart auf.

Anna versteht die folgenden 4 Anweisungen: `nachlinks()`; `nachoben()`; `nachrechts()`; `nachunten()`;



```
1 start();
2 nachrechts(); nachrechts();
3 nachoben(); nachoben();
4 nachlinks(); nachlinks(); nachlinks();
5 nachoben();
6
```

ausführen

>

Abbildung 8: Beispiel zu Grundlagen imperativer Programmierung

Abb. 9 zeigt eine Übungsaufgabe, bei der mit graphischen Hilfsmitteln ein Codestreifen für einfache Musikstücke erzeugt werden soll. Um diese Aufgabe lösen zu können, haben die Schülerinnen und Schüler in vorangegangenen Übungen Methoden zur Ton- und Grafikerzeugung kennen gelernt und angewandt. Ziel dieser Übung ist es, die Schleife als Kontrollstruktur zu thematisieren.

Musik veranschaulichen

[Bearbeiten] Java

Du kennst die Tonleiter und die Notenschrift aus dem Musikunterricht. Für Drehorgeln und Spieluhren wird hingegen eine Art Code in Form von Stifen verwendet, die auf einer Walze aufgebracht sind. Im Bild rechts sieht man das Innenleben einer Spieluhr.

In dieser Aufgabe sollst du versuchen, aus einem Lied als Zeichenkette wie:

```
String Lied = "E-4 E-4 E-2 E-4 E-4 E-2 E-4 G-4 C-4 "+
             "D-4 E-1 F-4 F-4 F-4 F-4 F-4 E-4 E-4 "+
             "E-8 E-8 E-4 D-4 D-4 E-4 D-2 G-2*";
```

eine Darstellung für eine zugehörige Walze einer Spieluhr zu erzeugen. Gleichzeitig kannst du das Lied auch mit Hilfe der Methoden, die wir letzte Woche kennengelernt haben, vorspielen lassen.



Spieluhr und Walze mit Stifen



```
1 | canvas.clear();
2 | String Lied = "E-4 E-4 E-2 E-4 E-4 E-2 E-4 G-4 C-4 "+
3 |             "D-4 E-1 F-4 F-4 F-4 F-4 F-4 E-4 E-4 "+
4 |             "E-8 E-8 E-4 D-4 D-4 E-4 D-2 G-2*";
5 |
6 | orgel = new Sound(19); // 19. MIDI-Gerät = Orgel
7 |
8 | String[] Noten = Lied.split(" ");
9 |
10 | int posX = 0;
11 | int posY = 0;
12 | for(int i=0; i < Noten.length; i++){
13 |     int key = 0;
14 |     if (Noten[i].charAt(0) == 'C') key = 60;
15 |     if (Noten[i].charAt(0) == 'D') key = 62;
16 |     if (Noten[i].charAt(0) == 'E') key = 64;
17 |     if (Noten[i].charAt(0) == 'F') key = 65;
18 |     if (Noten[i].charAt(0) == 'G') key = 67;
19 |     if (Noten[i].charAt(0) == 'A') key = 69;
20 |     if (Noten[i].charAt(0) == 'B') key = 71;
21 |     int dauer = 0;
22 |     if (Noten[i].charAt(2) == '1') dauer = 1000/1;
23 |     if (Noten[i].charAt(2) == '2') dauer = 1000/2;
24 |     if (Noten[i].charAt(2) == '4') dauer = 1000/4;
25 |     if (Noten[i].charAt(2) == '8') dauer = 1000/8;
26 |
27 |     orgel.play(key, dauer);
28 |     canvas.fillRect(posX, 90-(key-60)*8, dauer/20, 10);
29 |     posX = posX + dauer/20+5;
30 | }
31 |
```

speichern & ausführen

Abbildung 9: Beispiel zur Grafik- und Tonerzeugung

Literatur

- [Döb07] Beat Döbeli. Wiki und die starken Potenziale - Unterrichten mit Wikis als virtuellen Wandtafeln. *Zeitschrift Computer und Unterricht*, Web 2.0 und Schule Nr 66:39–41, 2007.
- [Her08] Heroku, Oktober 2008.
<http://heroku.com/>.
- [KH08] Christian Kohls und Simone Haug. Gemeinsam sind wir stark! - Kooperativer Wissenserwerb mit Wikis. *LOG IN*, 152:22–28, 2008.
- [Knu84] Donald E. Knuth. Literate Programming. *The Computer Journal*, vol.27:97–111, 1984.
- [Med08] MediaWiki, Oktober 2008.
<http://www.mediawiki.org>.
- [RH04] Raimond Reichert und Werner Hartmann. On the Learning in E-Learning. *Proceedings of EDMEDIA 2004*, World Conference on Education Multimedia, Hypermedia and Telecommunications June 23–26, Lugano, Swizerland:1590–1595, 2004.
- [Sch02] Rolf Schulmeister. Taxonomie der Interaktivität von Multimedia - Ein Beitrag zur aktuellen Metadaten-Diskussion. *it+ti*, Ausgabe 4:193–199, 2002.
- [Wag05] Christian Wagenknecht. Mediendidaktische Begleitung im Informatikunterricht mit SchRepo, SchemeNet und SchemeGrader. *Unterrichtskonzepte für informatische Bildung*, INFOS'05 - Praxisband:21–24, 2005.
- [WB05] Christian Wagenknecht und Veit Berger. Programmierparadigmen mit Scheme. *Unterrichtskonzepte für informatische Bildung*, INFOS'05:219–229, 2005.
- [ZUM08] ZUM Wiki, Oktober 2008.
<http://wiki.zum.de/>.

Von der ersten Entitätsklasse zum Webshop - Datenbanken in allgemein bildenden Schulen

Holger Rohland

Fakultät Informatik, Institut für SMT
Arbeitsgruppe Didaktik der Informatik / Lehrerbildung
TU Dresden
01062 Dresden
hr8@inf.tu-dresden.de

Abstract: Der vorliegende Beitrag beleuchtet die Möglichkeiten und Grenzen der Beschäftigung mit dem Thema Datenbanken in allgemein bildenden Schulen. Dabei wird deutlich, dass das Thema in allen Schularten in sinnvoller Weise in das Gesamtkonzept informatischer Bildung eingepasst werden kann. Es wird der Bogen gespannt von der Modellierung von Datenbasen und deren Auswertung mit einfachen Abfragen in der Sekundarstufe I bis hin zur Realisierung komplexer Projekte auf der Basis von XAMPP-Umgebungen in der gymnasialen Oberstufe. Der Beitrag schließt mit einigen Bemerkungen zu den Anforderungen an die Lehreraus- und -weiterbildung im Gebiet Datenbanken, weil ohne diese Ausbildung eine fachgerechte Gestaltung von Unterricht unmöglich ist.

1 Vorbemerkungen

Betrachtet man die Entwicklung informatischer Allgemeinbildung im Rückblick auf die vergangenen 30 Jahre, so fällt auf, dass das Thema Datenbanken in seiner heutigen Betrachtungsweise erst recht spät in den Fokus der Pädagogen, Fachdidaktiker und Lehrplangestalter gerückt ist. Lange Zeit galten Algorithmen und Datenstrukturen, Programmiersprachen und Hardwareaufbau als die schulrelevanten Teilgebiete der Informatik.

Dabei ist spätestens mit der breiten Einführung relationaler Datenbanken zu Beginn der 1970er Jahre klar geworden, dass Wissen rund um das Thema Datenbanken in der informatischen Bildung eine bedeutende Rolle spielen wird. Die rasche Verbreitung von Desktop Datenbankmanagementsystemen (DBMS) wie dBase, MS-Access oder Filemaker und der massenhafte Einsatz datenbankbasierter Komplettlösungen wie Warenwirtschafts- oder Kundenverwaltungssystemen in allen Lebensbereichen hat in den 1980er und 90er Jahren zu einem weiteren massiven Anstieg der Anzahl von Nutzern geführt, die in ihrer beruflichen Praxis mit Problemen rund um das Thema Datenbanken konfrontiert wurden.

Im Informatikunterricht allgemein bildender Schulen jedoch kam das Thema erst mit deutlicher Verspätung an. Noch 1994 konstatiert LOGIN im Einleitungstext zum Themenheft 'Datenbanken in der Schule', dass "das Unterrichtsthema 'Datenbanken' ... in der Schule bislang außerordentlich selten anzutreffen (ist)." [LO94]. Im Editorial des gleichen Heftes wird dann nochmals betont, dass sich Informatik-Grundausbildung fast nur mit Algorithmen befasst, die im Arbeitsspeicher laufen, während Hintergrundspeicher bzw. Datenbanksystem kaum vorkommen.

Selbst dann, wenn in den 80er Jahren ein Unterrichtsansatz unter das Label Datenbanken gestellt wurde, verbarg sich dahinter nicht der heutige Ansatz der Modellierung und Auswertung von Datenbasen. Das Thema wurde stattdessen als Alternative zur klassischen Dateiverwaltung positioniert, in der Regel wurden die Daten in einer einzigen Tabelle gespeichert und die Verwaltung dieser Daten erfolgte in der proprietären Sprache des Basissystems. Fast die Hälfte der Unterrichtszeit widmete sich Problemen des Datenschutzes (vgl. [KS86]) - eine wirkliche Annäherung an die Teildisziplin Datenbanken der Wissenschaft Informatik war damit definitiv nicht möglich.

Erst in den 90er Jahren hat das Thema mit seiner heutigen Schwerpunktsetzung schrittweisen Einzug in die Lehrpläne einzelner Bundesländer gehalten. So ist im Rahmenlehrplan Informatik des Bundeslandes Berlin von 1993 das Thema Datenbanken zum verbindlichen Lerninhalt des zweiten Unterrichtsjahres erklärt worden. Im Unterschied zu früheren Ansätzen spielten hier erstmalig Fragen der Modellierung (ER-Modell) und die Nutzung der Sprache SQL für das Generieren und Auswerten von relationalen Datenbanken eine Rolle.

Mit den 2008 veröffentlichten "Grundsätzen und Standards für die Informatik in der Schule - Bildungsstandards Informatik in der Sekundarstufe I" [BS08] und den seit 2004 gültigen "Einheitlichen Prüfungsanforderungen Informatik" [EPA04] für das Abitur hat man nun endgültig und für alle Bundesländer einheitlich Festlegungen zu Mindeststandards im Kontext des Wissensgebietes Datenbanken getroffen. So werden in den genannten Bildungsstandards für die Sekundarstufe I Kompetenzen formuliert, in denen von den Schülern erwartet wird, dass sie

- Strukturierungsmöglichkeiten von Daten zum Zusammenfassen gleichartiger und unterschiedlicher Elemente zu einer Einheit sowie grundlegende Operationen zum Zugriff auf die Bestandteile strukturierter Daten kennen und verwenden. ([BS08] S. 14)
- die Verwaltung und Speicherung großer Datenmengen mithilfe eines Datenmodells modellieren, einfache Datenmodelle in relationale Modelle umsetzen und diese mit einem Datenbanksystem realisieren können. ([BS08] S. 19)

Analysiert man die weiteren Ausführungen zu den genannten Bereichen, so werden im Kontext der zitierten Inhalte und Prozesse konkrete Forderungen an Schüler formuliert. Diese sollen sich mit Fragestellungen nach der Angemessenheit des verwendeten Informatiksystems, nach dem gewählten Datentyp, nach der Struktur der Daten oder der Vermeidung von Datenredundanzen auseinandersetzen. Der grundlegende Charakter dieser Fragen macht sichtbar, dass es sich bei den hier beschriebenen Sachverhalten tatsächlich um Mindestforderungen informatischer Bildung in der Sekundarstufe I handelt.

Zur Analyse der Anforderungen an die allgemeine Hochschulreife sollen die "Einheitlichen Prüfungsanforderungen Informatik" [EPA04] herangezogen werden. Hier werden für die Abiturprüfung folgende Forderungen an die Schüler gestellt:

Nachzuweisen sind "Kenntnisse der folgenden Modellierungstechniken" im Bereich "Datenmodellierung insbesondere: semantisches Datenmodell (Beschreibung der relevanten Objekte und ihrer Beziehungen, ER-Modell), logisches Datenmodell (z.B. relationales Datenmodell)".

Welche konkret erwarteten Kompetenzen hinter diesen Forderungen stehen, wird deutlich, wenn man die Schwerpunkte eines der dargestellten ausführlich kommentierten Beispiele betrachtet. So heißt es in Aufgabe 1.1 unter anderem: "Ergänzen Sie das Entity-Relationship-Modell ... definieren Sie auch die Attribute und Schlüssel", "Formulieren Sie ... SQL-Anweisungen für die aufgeführten Problemstellungen" und "Die Tabelle ... entspricht nicht den Normalisierungsregeln. Entwerfen Sie ein Relationenmodell, das diesen Mangel beseitigt."

Sichtbar wird das deutlich höhere Niveau der Anforderungen bei weitgehend identischen Inhaltsbereichen wie in der Sekundarstufe I. Dies entspricht dem spiralcurricularen Ansatz. Problematisch erscheinen diese Forderungen allerdings vor dem Hintergrund der existierenden Lehrpläne in den einzelnen Bundesländern. Ohne die Ausbildung der Grundlagen in einem eigenständigen Fach Informatik in der Sekundarstufe I sind die in der EPA gestellten Prüfungsanforderungen wohl schwerlich zu erreichen.

Der vorliegende Beitrag soll aufzeigen, wie eine durchgängige Ausbildung im Fachgebiet Datenbanken aussehen müsste, um Schüler letztlich in die Lage zu versetzen, die Anforderungen der Abiturprüfungen zu erfüllen. Dies erfolgt auf der Basis gesammelter Erfahrungen bei der Umsetzung der aktuellen sächsischen Lehrpläne sowie unter Berücksichtigung der Rahmenbedingungen der Aus- und Weiterbildung von Informatiklehrern an der Technischen Universität Dresden.

2 Datenbankgrundlagen in der Sekundarstufe I

Betrachtet man die Behandlung von Datenbanken in der Sekundarstufe I, so findet man im sächsischen Lehrplan für die Mittelschule [SN04] im Lernbereich 1 der Klassenstufe 9 ein ausgereiftes Konzept. Die fachgerechte Umsetzung der hier benannten Schwerpunkte sollte einen standardkonformen Kompetenzerwerb der Schüler ermöglichen. Im Mittelpunkt stehen dabei das Modellieren von Datenbanken und das Gewinnen neuer Informationen durch Abfragen.

2.1 Modellierung der Datenbasis

In den Bildungsstandards Informatik in der Sekundarstufe I [BS08] werden im Inhaltsbereich "Information und Daten" für Schüler der Jahrgangsstufen 5 bis 7 Kompetenzen bezüglich der Zusammenhänge von Informationen und Daten sowie hinsichtlich verschiedener Darstellungsformen für Daten formuliert. Es wird von Ihnen erwartet, dass sie die Begriffe Klasse, Objekt, Attribut und Attributwert kennen und sie in Anwendungssituationen benutzen.

Der sächsische Lehrplan der Klassen 7 und 8 wird dem mit dem Grundansatz der objektorientierten Betrachtung von Informatiksystemen gerecht. Die Schüler werden mit der Modellierung von Klassen und Objekten konfrontiert und lernen verschiedene Notationsformen kennen.

Auf dieser Basis ist ein ausführliches Konzept zum Thema Modellierung von Datenbasen vorgestellt worden (vgl. [NR07]). Zusammenfassend lassen sich aus diesem Unterrichtsentwurf zwei grundlegende Aspekte herausarbeiten: Einerseits ist zu klären, ob Klassendiagramme, Mindmaps oder ER-Diagramme (ERD) zum Einsatz kommen sollten und andererseits ist zu analysieren, wie Schülern die Bestimmung der Kardinalität von Beziehungen vereinfacht werden kann.

Erste Erfahrungen bei der Umsetzung der dargestellten Ansätze zur Modellierung haben gezeigt, dass sowohl der ERD-Ansatz als auch das Mindmap nicht zu unterschätzende Probleme mit sich bringen. Im ER-Diagramm liegen diese vor allem in der abweichenden Form der Modelldarstellung gegenüber den bis dahin im Unterricht behandelten Formen sowie in der fehlenden zweiten Leserichtung im Kontext der korrekten Bestimmung des Beziehungstyps.

Mindmaps haben den Vorzug, dass diese den Schülern aus anderen Fächern geläufig sind. Werden Mindmaps aber im Informatikunterricht der Klasse 7/8 verwendet, dann vor allem zur Darstellung von Klassen. Dabei steht in der Regel der Klassenname in der Wurzel des Mindmaps, an den Ästen werden die Attribute dargestellt. Dass im hier gewählten Ansatz plötzlich der Name der Datenbasis in der Wurzel erscheint und an den Ästen Klassen stehen, sorgt nicht nur bei leistungsschwächeren Schülern für Verwirrung.

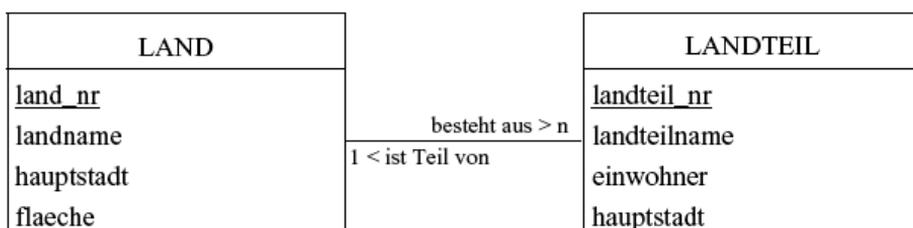
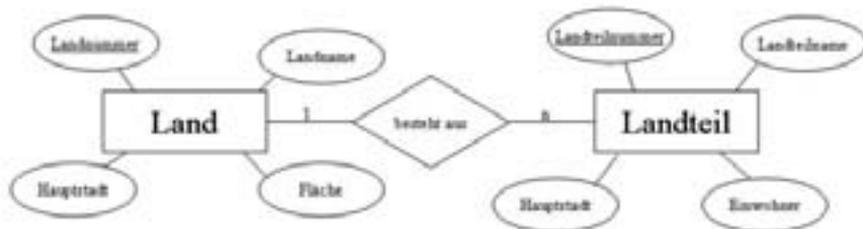


Abbildung 1: konkurrierende Ansätze zur Modellierung

In Anbetracht dieser Probleme und angesichts der Tatsache, dass in zunehmendem Maße auch in der Fachwissenschaft (nicht zuletzt befördert durch den Einsatz von CASE-Tools) Klassendiagramme zum Einsatz kommen, ist aus heutiger Sicht die Verwendung dieser Notationsform auch im Unterricht zu präferieren. Im Vergleich zum ERD lässt sich im dargestellten Beispiel sehr gut erkennen, dass die Beschreibung der Beziehung zwischen den Klassen mit "1 Land besteht aus n Landteilen" und "1 Landteil ist Teil von 1 Land" nicht nur eindeutiger ist, sondern zugleich das muttersprachliche Prinzip im Unterricht besser unterstützt.

Für die Bestimmung der Kardinalität von Beziehungen hat sich der in [NR07] dargestellte formalisierte Ansatz grundsätzlich bewährt. Allerdings sei an dieser Stelle darauf hingewiesen, dass die Wahl der Beispiele ganz wesentlich das Verständnis für diese Aufgabe beeinflusst. So gibt es zahlreiche Beispiele, in denen die Kardinalität nicht immer eindeutig bestimmt werden kann, weil z.B. erst konkrete juristische oder betriebliche Festlegungen dies tatsächlich möglich machen würde.

Problematisch wird es vor allem dann, wenn der zeitliche Aspekt ins Spiel kommt. Schüler haben beispielsweise immer wieder Schwierigkeiten, die Beziehung "Leser – leiht aus – Buch" als $n : m$ Beziehung zu identifizieren, weil der Satz "1 Buch wird von n Lesern ausgeliehen" den Widerspruch provoziert, dass ein Buch nicht *gleichzeitig* von mehreren Lesern ausgeliehen werden kann. Beispiele ohne zeitlichen Bezug wie oben dargestellt sind daher für den Unterricht in der Sekundarstufe I eindeutig besser geeignet.

2.2 Modellierung von Abfragen

Auch die Modellierung von Abfragen ist in [NR07] ausführlich besprochen worden. Das vorgestellte Modell folgt dabei der Intention, auch im Bereich Abfragen das Planen (auf Papier) vom eigentlichen Implementieren (mit dem DBMS) zu trennen.

<Datenfeldliste>:	
<Tabellenliste>:	
<Bedingungen>:	
<Eigenschaften>:	
<Gruppierungen>:	
<Einschränkungen>:	
<Sortierungen>:	

Abbildung 2: Modellierungsansatz für Abfragen in der Sekundarstufe I

In der Praxis hat sich dieser Ansatz bewährt. Er ist für Schüler der Sekundarstufe I leicht verständlich, bietet darüber hinaus aber auch hinreichende Differenzierungsansätze. So ist es beispielsweise möglich, die Modellierung dadurch zu vereinfachen, dass in der Aufgabenstellung von vornherein nur die Zeilen erscheinen, die für die Lösung des Problems relevant sind. Durch Angabe aller Zeilen erhöht sich der Schwierigkeitsgrad automatisch. Die wahlweise Übertragung des verbalen Ansatzes in den QBE-Bereich eines DBMS oder dessen Transformation in die Sprache SQL stellen weitere Differenzierungsvarianten dar.

Zusammenfassend lässt sich feststellen, dass die in den Bildungsstandards Informatik für den mittleren Schulabschluss geforderten Kompetenzen im Umfeld des Themas Datenbanken bei qualifizierter Erteilung von Informatikunterricht durchaus erreichbar sind. Sowohl die Modellierung als auch die Auswertung von Datenbeständen mittels Abfragen lassen sich didaktisch so weit reduzieren, dass die Mehrzahl der Schüler in der Lage sein sollte, die in den Standards geforderten Kompetenzen zu erwerben.

3 Datenbankprojekte in der Sekundarstufe II

Im Unterschied zum deduktiven Ansatz in der Sekundarstufe I, bei dem grundlegendes Wissen fachsystematisch vermittelt wird, empfiehlt sich für die gymnasiale Oberstufe ein induktiver Ansatz. Während bei ersterem der Instruktion durch den Lehrer das Primat zukommt, sollte hier ein handlungsorientiertes Vorgehen favorisiert werden. Interessante Impulse liefert dabei der Lernfeldansatz in der beruflichen Bildung. (vgl. [LR03])

Auch wenn in der allgemein bildenden Schule der Bezug zum konkreten beruflichen Handlungsfeld fehlt, so kann doch eine Vielzahl der Überlegungen zur Gestaltung von Lernfeldern auf den Unterricht in der Sekundarstufe II übertragen werden. Im Folgenden soll eine mögliche Variante der Behandlung des Lernbereichs Datenbanken im Grundkurs der Abiturstufe vorgestellt werden.

An den Anfang eines solchen Szenarios gehört ein für den Schüler bedeutsames, möglichst authentisches Problem. Gerade im vorliegenden Fall ist es naheliegend, auf die Erfahrungen der Schüler mit unterschiedlichsten Webpräsenzen zurückzugreifen. Es ist leicht möglich, den Schülern deutlich zu machen, dass alle von ihnen häufig frequentierten Webangebote, seien es Communities, Auktionsplattformen oder Webshops, nur so funktionieren können, dass die letztlich angezeigte Webseite zum Zeitpunkt des Aufrufs dynamisch erzeugt wird.

Im weiteren Verlauf macht es sich dann erforderlich zu klären, was im Unterschied zu statischen Webseiten, die der Schüler sinnvoller Weise zuvor schon mittels einfacher HTML-Dokumente erstellt hat, beim Generieren von dynamischen Seiten notwendig ist. Dabei werden die Schüler in Anwendung vorhandenen Grundlagenwissens aus der Sekundarstufe I weitgehend selbständig folgende Problemkreise herausarbeiten können:

- man benötigt eine klassische HTML-Seite für die statischen Elemente der Website
- es bedarf einer Datenbank, aus der die aktuellen Daten gelesen werden können
- es ist eine Programmiersprache notwendig, mit deren Hilfe man auf die Inhalte der Datenbank zugreifen kann

Vorausgesetzt, die Schüler verfügen in allen drei Bereichen über die in den "Standards" geforderten Kompetenzen, so kann der weitere Unterricht in hohem Maße schülerzentriert erfolgen. Eigene Unterrichtserfahrungen haben gezeigt, dass sich Schüler weitgehend selbstständig in eine neue Programmiersprache (z.B. PHP) einarbeiten können. Hierfür genügte eine einfache tabellarische Gegenüberstellung der Syntax einer bekannten Programmiersprache (z.B. Pascal) mit der hier neu zu erlernenden, welche natürlich auf dem gleichen Grundparadigma basieren muss.

Im weiteren Verlauf des Unterrichts hat es sich als völlig ausreichend erwiesen, die wenigen spezifischen Funktionen zu erläutern, die den Datenbankzugriff von PHP auf eine MySQL-Datenbank steuern. Den "Rest" konnte die Mehrzahl der Schüler durch Analogieschlüsse zu Problemen der klassischen Programmierung selbstständig bewältigen.

Während also im Problemkreis Programmierung instruktionale Unterstützung eher von untergeordneter Bedeutung war, gestaltete sich deren Notwendigkeit bezüglich des Zugriffs auf die Datenbank etwas komplexer. Dies betraf weniger die Modellierung der Datenbasis, hier war das vorhandene Vorwissen völlig ausreichend, sondern vielmehr die Erzeugung von Datenbankabfragen mittels der Sprache SQL. An dieser Stelle hat sich fachsystematisches Vorgehen bewährt.

Dabei ist es im Sinne des angestrebten situierten Ansatzes durchaus sinnvoll, zunächst mit einfachen, intuitiv zu verstehenden SELECT-Anweisungen erst funktionierende dynamische Webseiten zu erzeugen. Im weiteren Verlauf erwies es sich jedoch als zweckmäßig, den Gesamtkomplex SQL in strukturierter Weise zu behandeln. Erst aus einer Gesamtsicht der Schüler auf die Gruppe von SQL-DML-Statements (SELECT, UPDATE, DELETE und INSERT) war es den Schülern möglich, den einzelnen Funktionen des Webshops wie "Artikel in den Warenkorb legen" oder "Bestellmenge aktualisieren" die richtigen und syntaktisch korrekten SQL-Anweisungen zuzuordnen.

Leider erlaubt es der zur Verfügung stehende zeitliche Rahmen den Schülern an allgemein bildenden Schulen im Rahmen des regulären Unterrichts nicht, das Wissen an dieser Stelle durch selbständige Arbeit an größeren Projekten zu vertiefen. Dazu muss man ihnen ein wesentlich größeres Zeitvolumen einräumen, wie dies z.B. im Rahmen einer Berufsausbildung, einer besonderen Lernleistung (BELL) oder einer Jahresarbeit im Schülerrechenzentrum der TU Dresden (vgl. [HTU03]) geschieht.

Vorliegende Ergebnisse solcher Projekte berechtigen zu der Aussage, dass der dargestellte Ansatz Schülern nicht nur erlaubt Lernziele zu erreichen, die zur Erfüllung der einheitlichen Prüfungsanforderungen ausreichen, sondern darüber hinaus tatsächlich anwendbares Wissen im Wissensgebiet Datenbanken zu erwerben.

Abschließend soll an dieser Stelle hervorgehoben werden, dass beim dargestellten Ansatz dem Zusammenspiel von Instruktion und Konstruktion eine ganz besondere Bedeutung zukommt. Es hat sich in der Praxis gezeigt, dass die Schüler neben Phasen selbständigen Erprobens von Problemlösungen auch immer wieder gezielte Anleitung und wohlstrukturierte Vermittlung von Grundlagen benötigen. Damit wurde bestätigt, dass "das Lernen unter multiplen Perspektiven zwar das Potenzial (hat), einen anwendungsbezogenen Wissenserwerb zu fördern: doch dieses Potenzial kommt nur dann zur Entfaltung, wenn die Komplexität zusätzlich mit instruktionaler Unterstützung kombiniert wird." [RM06].

4 Anforderungen an die Lehreraus- und Weiterbildung

Im Folgenden soll abschließend deutlich gemacht werden, dass gerade im betrachteten Teilgebiet der Informatik gut ausgebildete Lehrer ein zwingendes Erfordernis sind. Erfahrungen zeigen, dass ohne solide wissenschaftliche Grundlagen der Versuch dieses Thema zu unterrichten nahezu zwangsläufig zu krassen Fehlleistungen führt.

Im Gespräch mit Informatiklehrern, die seit mehr als 10 Jahren das Fach lehren, aber erst jetzt im Rahmen eines berufsbegleitenden Studiums die notwendige Qualifikation nachholen, sind insbesondere folgende Probleme zutage getreten:

- Das Thema Datenbanken wird wie Textverarbeitung oder Tabellenkalkulation behandelt (im günstigen Fall unter Beachtung objektorientierter Ansätze).
- Es erfolgt keine Modellierung im Sinne von Datenbanken, eine Tabelle wird als Datenbank verstanden, womit die Behandlung von Beziehungen zwischen Tabellen komplett entfällt.
- Das Definieren von Datenfeldern und das Eingeben und Bearbeiten der Tabelleninhalte werden als die wichtigsten Aufgaben angesehen, bestenfalls werden Daten in der Tabellenansicht gefiltert, anstatt Abfragen zu modellieren und zu implementieren.

Ziel der Aus- und Weiterbildung von Informatiklehrern muss es deshalb sein, das Teilgebiet "Datenbanken" sowohl in der Breite als exemplarisch auch in der Tiefe zu behandeln. Dem kann man mit einem Ansatz gerecht werden, der in einer handlungsorientierten Erstbegegnung, einer fachwissenschaftlichen Grundlagenvorlesung und einer fachdidaktischen Vertiefung besteht.

Im Studiengang Lehramtsbezogener Bachelor "Allgemeinbildende Schulen" und Lehramtsbezogener Bachelor "Berufliche Bildung" an der Technischen Universität Dresden wird dem mit drei aufeinander folgenden Modulen Rechnung getragen. In einem ersten Modul erfolgt ein praxisorientierter Einstieg in das Thema Datenbanken aus Anwendersicht. Neben Modellierungsfragen geht es deshalb auch um Kompetenzerwerb bezüglich der Nutzung eines DBMS.

Die Studentinnen und Studenten sollen in die Lage versetzt werden, relationale Datenbanken zu planen, zu implementieren und zu nutzen. Dabei wird das allgemeine Vorgehen erläutert und praktisch am Datenbankmanagementsystem ACCESS realisiert. Wesentlicher Schwerpunkt ist dabei die didaktische Reduktion des komplexen Problems Datenbanken für die Nutzung im Unterricht.

Dem schließt sich ein zweites Modul an, welches die Studierenden der Lehramtsbezogener Bachelorstudiengänge gemeinsam mit allen Informatikstudenten absolvieren. Dabei handelt es sich um das Grundlagenmodul zum Wissenschaftsgebiet Datenbanken. Dessen Lehrziel besteht darin, den Studierenden Kenntnisse der Datenbanktheorie zu vermitteln und sie in die Lage zu versetzen, Anwendungsprobleme praktisch zu lösen.

Schwerpunkte sind einerseits das Entity-Relationship-Modell, das relationale Datenmodell einschließlich der Entwurfstheorie relationaler Datenbanken und das XML-Datenmodell sowie andererseits Themen zur Realisierung von Datenbanksystemen, der Fehlerbehandlung und der Anfrageverarbeitung in Datenbanksystemen. Die Studierenden erwerben praktische Erfahrungen mit der relationalen Datenbanksprache SQL und der XML-Anfragesprache XQUERY.

Im dritten Modul lernen die Studentinnen und Studenten das Erzeugen von dynamischen Webseiten als ganz wesentliches aktuelles Beispiel für die Anwendung von Datenbanken kennen. Nachdem in einem ersten Teil dieses Moduls Grundlagen der Sprache PHP erläutert und für einfache Beispiele und webtypischen Anwendungen genutzt werden, spielt der Zugriff auf MySQL-Datenbanken die zentrale Rolle im weiteren Verlauf dieses Moduls. Dabei werden die in der Grundlagenlehrveranstaltung erworbenen Kompetenzen vertieft, indem Modellierung und SQL-Praxis in komplexen Aufgabenstellungen wie der Generierung eines datenbankbasierten Wissenstests zusammenfließen.

Das zuerst wie das letztgenannte Modul besuchen nur Studierende der Lehramtsbezogenen Bachelorstudiengänge. Dadurch ist es möglich, fachdidaktische Aspekte, wie die notwendige didaktische Reduzierung bei der Behandlung des Themas in der Sekundarstufe I, bereits hier explizit zu thematisieren. In den speziellen Modulen zur Fachdidaktik können diese Aspekte dann aufgegriffen und vertieft werden.

Neben den Bachelorstudiengängen spielen nach wie vor die berufsbegleitende Weiterbildung von Informatiklehrern und deren ständige Fortbildung eine wichtige Rolle. Vor allem die gegenwärtig zu verzeichnenden Studentenzahlen in der Erstausbildung machen es auf lange Sicht erforderlich, Lehrern im Berufsleben die Möglichkeit zu geben, sich als Fachlehrer für Informatik weiterzubilden.

Seit 1992 gibt es dafür die berufsbegleitende Weiterbildung im Fach Informatik - ein 4- bzw. 6-semestriges Studium mit einem Präsenztage pro Woche. Auf Initiative der GI-Fachgruppe "Informatische Bildung in Sachsen und Thüringen" nehmen seit dem Sommersemester 2008 vermehrt Kollegen die Möglichkeit wahr, an regulären Lehrveranstaltungen der TU Dresden teilzunehmen und ein Zertifikat zu erwerben, ganz besonders im Lehrgebiet Datenbanken!

Zusammenfassend lässt sich festhalten, dass mit den Bildungsstandards Informatik in der Sekundarstufe I wie auch mit den Einheitlichen Prüfungsanforderungen Informatik für die Abiturprüfung eine klare Schwerpunktsetzung bezüglich des Themas Datenbanken erfolgt ist. Das hier vorgestellte durchgängige Unterrichtskonzept zeigt, dass es möglich ist, die darin geforderten Kompetenzen bei den Schülern auszuprägen.

Notwendig dafür ist, dass solide Grundlagen im Sinne fachsystematischen Fakten- und Regelwissens kombiniert werden mit handlungsorientierten Lernprozessen, bei denen den Schülern die hohe Relevanz des Themas Datenbanken bewusst wird. Das wiederum wird nur dort gelingen, wo für dieses Fach ausgebildete Lehrer einen durchgängigen Informatikunterricht von Klasse 7 bis 12 nach Lehrplänen gestalten können, die für die Umsetzung dieses Konzepts genügend Freiräume bieten.

Literaturverzeichnis

- [BS08] Arbeitskreis Bildungsstandards: Grundsätze und Standards für die Informatik in der Schule - Bildungsstandards Informatik in der Sekundarstufe I. Beilage zu: LOGIN 150/151, Berlin, 2008.
- [EPA04] Kultusministerkonferenz: Einheitliche Prüfungsanforderungen Informatik. http://www.kmk.org/fileadmin/pdf/PresseUndAktuelles/Beschluesse_Veroeffentlichungen/allg_Schulwesen/EPA-Informatik.pdf [Stand: 12.02.2009]
- [FKN06] Fischer, H.; Knapp, T.; Neupert, H.: Grundlagen der Informatik II, Oldenbourg Schulbuchverlag GmbH, München, 2006
- [FKN07] Fischer, H.; Knapp, T.; Neupert, H.: Grundlagen der Informatik II – Lehrermaterialien CD-ROM,, Oldenbourg Schulbuchverlag GmbH, München, 2007
- [HTU03] Steffi Heinicke, Bettina Timmermann, Michael Unger; Konzepte für die Begabtenförderung auf dem Gebiet der Informatik und ihre Umsetzung am Schülerrechenzentrum Dresden; in: P. Hubwieser (Hrsg.) Informatische Fachkonzepte im Unterricht, Lecture Notes in Informatics Vol. P 32, Bonn, 2003
- [KS86] Käberich, G.; Steigerwald, F.: Schüler arbeiten mit einer Datenbank, Metzler & Teubner, Stuttgart, 1986
- [LO94] LOGIN 2/1994, Berlin, 1994.
- [LR03] Linke, H.; Rohland, H.: Lernfeldorientierter Ansatz in der Berufsausbildung - Chancen und Risiken; in: P. Hubwieser (Hrsg.) Informatische Fachkonzepte im Unterricht, Lecture Notes in Informatics Vol. P 32, Bonn, 2003
- [NR07] Neupert, H.; Rohland, H.: Modellierung der Datenbasis von Datenbanken. In: LOGIN 146/147, Berlin, 2007; S. 27-35.
- [RM06] Reinmann, G.; Mandl, H.: Unterrichten und Lernumgebungen gestalten. In: A. Krapp & B. Weidenmann (Hrsg.), Pädagogische Psychologie (S.613-658), Beltz Verlag, Weinheim und Basel, 2006
- [SN04] Sächsisches Ministerium für Kultus (Hrsg.): Lehrplan Informatik für die Mittelschule. http://www.sachsen-macht-schule.de/apps/lehrplandb/downloads/lehrplaene/lp_ms_informatik.pdf [Stand: 12.02.2009]

Mit iLearnIT.ch spielerisch das Interesse an Informatik wecken

Beat Döbeli Honegger, André Frey, Philippe Braxmeier

Institut für Medien und Schule (IMS)
Pädagogische Hochschule Zentralschweiz - Schwyz
Zaystrasse 42
6410 Goldau
beat.doebeli@phz.ch
andre.frey@phz.ch
philippe.braxmeier@phz.ch

Abstract: Im Rahmen des Schweizer Jahres der Informatik wurde mit der Entwicklung des kostenlosen Online-Webangebotes iLearnIT.ch begonnen, das Kindern und Jugendlichen die Informatik und ihre Konzepte spielerisch näher bringen will. Der Beitrag dokumentiert die hinter iLearnIT.ch stehenden didaktischen Überlegungen, die bei der Umsetzung angetroffenen Herausforderungen und wirft einen Blick auf die weitere Entwicklung.

1 Ausgangslage

Computer, Handy, Spielkonsole, MP3-Player, Internet, aber auch berührungslose Ticketkontrolle am Skilift: Kinder und Jugendliche wachsen heute in der Informationsgesellschaft auf und nutzen die allgegenwärtigen ICT-Produkte und -Dienstleistungen ohne Scheu oder Vorbehalte. So besitzen bereits 66% der 10 bis 11-jährigen Kinder in Deutschland ein Mobiltelefon, bei den 12- bis 13-jährigen sind es sogar 86% [MPFS08].

Dieser alltägliche Umgang kontrastiert jedoch oft mit der Unkenntnis der dahinter steckenden Konzepte der Informatik. Weder werden Informatikkenntnisse in der Schweiz – mit wenigen Ausnahmen – in der Schule vermittelt, noch bestehen für Kinder und Jugendliche Anreize oder Möglichkeiten, sich dieses Wissen ausserhalb der Schule anzueignen. Informatik erhält so bei Kindern und Jugendlichen nicht das Bild eines interessanten Fachgebiets, sondern eines Alltagsthemas, das auf Versuch und Irrtum aufbaut und bei dem es fast zum guten Ton gehört, die Konzepte dahinter nicht zu verstehen.

Dieses Bild der Informatik wirkt sich negativ auf die Berufswahl von Jugendlichen aus. Zuwenig Jugendliche wählen in der Schweiz ein Informatikstudium oder eine Informatik-Berufslehre, um den Bedarf der Wirtschaft an Informatikerinnen und Informatikern zu decken [Zeh08]. Zwischen 2001 und 2008 ist die Zahl der Neustudierenden an Schweizer Hochschulen in Informatik und Wirtschaftsinformatik um 60%, an Fachhochschulen um ca. 30% eingebrochen [Zeh08]. Dieser Einbruch ist zwar auch in anderen Ländern zu beobachten, ist aber in der Schweiz besonders ausgeprägt. Auch in einem weiteren Punkt nimmt die Schweiz eine Sonderstellung ein: Nirgends sonst ist der Frauenanteil in Informatikausbildungen und –berufen so gering wie in der Schweiz. Trotz gezielter Förderprogramme konnte er nicht über 15% gehoben werden.

Diese Situation nahmen die Schweizer Informatikverbände zum Anlass, ein *Schweizer Jahr der Informatik* auszurufen. Ähnlich wie im Jahr 2006 in Deutschland, so sollten im Jahr 2008 Publikationen, Aktivitäten und Veranstaltungen das Image der Informatik in der Bevölkerung, aber insbesondere bei Kindern und Jugendlichen fördern. Finanziert wurde das Schweizer Jahr der Informatik von der Privatwirtschaft und der Hasler Stiftung.

Zur Förderung des Informatikwissens von Kindern und Jugendlichen sind gute Lehrmittel rar. Zwar existieren Lehrmittel zu Nutzung von ICT (Anwendungswissen), produktorientierte Informatikeinführungen oder aber eher abstrakte, nicht an die Lebenswelt von Kindern und Jugendlichen angepasste Informatiklehrmittel für Studierende. Die Hasler Stiftung teilte diese Einschätzung und unterstützt darum die Entwicklung eines stufengerechten Online-Lernmittels durch das Institut für Medien und Schule (IMS) der Pädagogischen Hochschule Zentralschweiz (PHZ) massgeblich. Das Projekt iLearnIT.ch wurde im Schweizer Informatikjahr gestartet, ist aber von Anfang an darauf ausgelegt, darüber hinaus weiter entwickelt zu werden und langfristig Wirkung zu erzielen.

2 Ziele

Das primäre Zielpublikum von iLearnIT.ch sind Kinder und Jugendliche. iLearnIT.ch soll Kindern und Jugendlichen die Möglichkeit geben, sich in der Schule aber auch ausserchulisch mit informatischen Themen zu beschäftigen. Damit werden zwei Ziele verfolgt:

1. **Allgemeinbildendes Ziel:** Kinder und Jugendliche lernen Möglichkeiten, Konzepte und Gedankenwelt der Informatik als eine der Basiswissenschaften der heutigen Informationsgesellschaft kennen. Damit soll das Wissen um die Themen und Konzepte der Informatik als Teil der Allgemeinbildung gefördert werden.

2. **Berufsbildendes Ziel:** Aus diesem Kontakt mit informatischen Themen kann unter Umständen ein Interesse an Informatik geweckt und vergrössert werden, das in der Folge auch die Studien- und/oder Berufswahl beeinflussen kann. Ziel ist hier die Erhöhung der Zahl der Jugendlichen, die ein Informatikstudium aufnehmen oder eine Informatiklehre beginnen.

Das für Kinder und Jugendliche genannte allgemeinbildende Ziel lässt sich auch auf **Eltern und Lehrpersonen** ausdehnen. Im Vergleich zu anderen Themen der Allgemeinbildung ist es eine Eigenheit des jungen Themas Informatik, dass Eltern und Lehrpersonen ohne Informatik aufgewachsen und zur Schule gegangen sind. Somit ist die Gefahr gross, dass Erwachsenen das entsprechende Wissen und vor allem das Bewusstsein für das Thema Informatik fehlen.

Dieses fehlende Bewusstsein ist der Hauptgrund, warum mit iLearnIT.ch ein drittes, **heimliches Zielpublikum** angesprochen werden soll: EntscheidungsträgerInnen in Politik und Bildungswesen. So sind in der Schweizerischen Politik informatiknahe Berufsgruppen im Vergleich zum Anteil an entsprechenden Schweizer Arbeitsplätzen untervertreten¹. Dies birgt die Gefahr, dass Fragen der Informatikbildung bei politischen Entscheiden nicht das notwendige Gewicht beigemessen wird. Mit iLearnIT.ch soll das Bewusstsein von Informatik als Bestandteil der Allgemeinbildung in der Informationsgesellschaft bei EntscheidungsträgerInnen erhöht werden.

Neben der Absicht, Bewusstsein und Wissen im Bereich Informatik zu fördern soll mit iLearnIT.ch auch die Verfügbarkeit von bisher weitgehend fehlendem fachlich und didaktisch hochstehendem Unterrichtsmaterial für die anvisierten Schulstufen erhöht werden.

3 Gestaltungsprinzipien und ihre Umsetzung

Um die oben genannten Ziele zu erreichen, wird bei der Planung und Umsetzung auf folgende Prinzipien geachtet:

- Fokussierung auf Konzeptwissen
- Durchdachte Themenwahl
- Stufengerechtigkeit / Spiralcurriculum
- Gender Mainstreaming
- Lebensweltbezug
- Entdeckendes Lernen
- computer science unplugged
- Mehrsprachigkeit
- Scratch als Programmierumgebung

¹ Gemäss Statistik der Parlamentsdienste (<http://www.parlament.ch/D/dokumentation/in-statistiken-tabellen/Seiten/in-st-berufe.aspx>) sitzen im Februar 2009 im 200-köpfigen Nationalrat nur 5 Personen der Berufsgruppe „Technik und Informatik“ während beispielsweise die Land- und Forstwirtschaft mit 24 Personen vertreten ist.

3.1 Fokussierung auf Konzeptwissen

In der Informatik kann grob zwischen langlebigem Konzeptwissen und eher kurzlebigen Produktwissen unterschieden werden [HNR06]. Bei iLearnIT.ch ergibt sich die Fokussierung auf Konzeptwissen direkt aus den Projektzielen: Kinder und Jugendliche sollen *Konzepte* der Informatik näher gebracht werden (da sie von den Produkten der Informatik ja schon umgeben sind). Das Weglassen von Produktwissen erhöht aber auch die zeitliche Gültigkeit eines Informatik-Lernmittels. In iLearnIT.ch sind deshalb mit Ausnahme der Programmiersprache Scratch keine produktspezifischen Informationen zu finden.

3.2 Durchdachte Themenwahl

Da das Aufzeigen des breiten thematischen Spektrums der Informatik eines der Hauptanliegen von iLearnIT.ch darstellt, kommt der Themenwahl eine wichtige Bedeutung zu. Doch die Definition von Informatik ist nicht einfach. Es fehlt im Vergleich zu anderen Wissenschaften ein historisch gewachsener Konsens, was zur Informatik gehört. Zudem ist bereits die Definition von Informatik als *Wissenschaft* nur eine von mehreren möglichen Perspektiven. Entsprechend existieren zahlreiche unterschiedliche Definitionen des Begriffs Informatik (siehe z.B. [Dö09]). Sowohl im deutschen als auch im Schweizerischen Jahr der Informatik wurden deshalb Broschüren unter dem Titel „Was ist Informatik?“ veröffentlicht ([BCM06] und [SG08]).

Bei ersten Überlegungen zum thematischen Spektrum von iLearnIT.ch wurden vier mögliche Perspektiven identifiziert:

1. Akademisch normativer Zugang I: Fundamentale Ideen der Informatik

In der deutschsprachigen Informatik-Didaktik hat sich der von Schwill begründete Ansatz der fundamentalen Ideen der Informatik weitgehend durchgesetzt [Sch93]. Ausgehend von Überlegungen von Bruner [Bru60] definiert Schwill eine Idee dann als fundamental, wenn sie

- seit längerem und auch zukünftig absehbar relevant ist (Zeitkriterium),
- in verschiedenen Bereichen anwendbar oder erkennbar ist (Horizontalkriterium),
- auf jedem intellektuellen Niveau demonstrierbar und vermittelbar ist (Vertikal-kriterium),
- sowie einen Bezug zur Lebenswelt der Lernenden (Sinnkriterium) hat.

2. Akademisch normativer Zugang II: Great principles of computer science

Peter Denning identifizierte 2003 ursprünglich fünf Grundprinzipien der Informatik [De03], die inzwischen auf folgende sieben Prinzipien erweitert worden sind [De07]: *Computation, Communication, Coordination, Recollection, Automation, Evaluation, Design*.

3. Akademisch faktischer Zugang: Informatik-Institute von Hochschulen

Eine weitere Möglichkeit, aktuelle Themen der Informatik zu definieren, besteht darin, von einer oder mehreren Hochschulen die Liste der Institute und ihrer thematischen Ausrichtung zu konsultieren.

4. Berufsbildend faktischer Zugang: Modullandschaft I-CH

Während die bisherigen Perspektiven alle akademisch ausgerichtet waren, lässt sich Informatik auch über die existierenden Ausbildungsmodule des Berufsbildungsfeldes Informatik definieren. Für die Schweiz bietet sich hier der Modulbaukasten von I-CH an, der Module verschiedener Informatikberufe kompetenzorientiert beschreibt.

Im Rahmen eines Themenfindungsworkshops haben ExpertInnen aller Schulstufen, Informatikfachleute aus verschiedenen Branchen sowie der Projektleiter des Projekts *Einstieg Informatik* [PKH07], Wolfgang Pohl, die vier oben genannten vier Perspektiven geprüft und mögliche Themen diskutiert. Daraus entstand die in Tabelle 1 abgebildete Struktur mit zwölf Themen. Es ist geplant, von den sieben Prinzipien von Denning deren sechs für je zwei Themen zu verwenden. Bei zwölf Themen wird somit jedes verwendete Prinzip von zwei Seiten her betrachtet.

	Thema	←	Great Principle of Computing	→	Thema	
1	Codierung	öffnen	<i>communication</i>	schliessen	Kryptographie	7
2	Algorithmen	Möglichkeiten	<i>computation</i>	Grenzen	Grenzen der Berechenbarkeit	8
3	Suchen / Sortieren	Gegebenes wieder finden	<i>recollection</i>	Neues sinnvoll ablegen	Persönliches Wissensmanagement	9
4	Prozesskoordination		<i>coordination</i>		Netzwerke / Internet	10
5	Usability	Anwendungssicht	<i>design</i>	Entwicklungssicht	Software Engineering	11
6	Realität / Virtualität	human	<i>automation</i>	artificial	Künstliche Intelligenz	12

Tabelle 1: Geplante Themenstruktur von iLearnIT.ch

3.3 Stufengerechtigkeit / Spiralcurriculum

Im Projekt iLearnIT.ch wird jedes Thema in den drei Schwierigkeitsstufen *leicht*, *mittel* und *schwer* präsentiert. Dies ist einerseits eine Folge des breiten anvisierten Alterspektrums, das altersgerechte Zugänge erfordert. Andererseits wird damit der praktische Beweis erbracht, dass die behandelten Themen das für fundamentale Ideen der Informatik [Sch93] erforderliche *Vertikalkriterium* erfüllen. Mit dem damit umgesetzten *Spiralcurriculum* [Bru60] kann Entscheidungsträgerinnen und -trägern in Politik und Bildungswesen gezeigt werden, dass informatische Bildung ein solides, didaktisch fundiertes Gerüst aufweist.

Die drei Niveaus sind auf die drei Schulstufen *Primarschule* (ab 5. Klasse), *Sekundarstufe I* (7.-9. Klasse) und *Sekundarstufe II* (10.-12. Klasse) ausgerichtet. Um diesen verschiedenen Schul- und Alterstufen gerecht zu werden, sind verwendete Beispiele, Sprache und Layout / Navigation der Lektionen den entsprechenden Schulstufen angepasst. So werden beispielsweise die Lernenden auf dem Niveau *leicht* per Du angesprochen und sowohl Webseiten als auch Sätze sind möglichst kurz gehalten. Auf der Stufe *schwer* wird auf eine direkte Ansprache der Lernenden verzichtet, dafür werden Nebensätze sowie Fach- und Fremdwörter verwendet.

3.4 Gender Mainstreaming

Da in der Schweiz der Frauenanteil in Informatikberufen und Informatikausbildungsgängen sowohl absolut gesehen als auch im europäischen Vergleich sehr tief ist, wird im Projekt iLearnIT.ch dem Thema Gender besondere Beachtung geschenkt und entsprechende Empfehlungen zu Gender Mainstreaming bei der Lehrmittelerstellung (Ry04) werden berücksichtigt. Mädchen und junge Frauen sollen sowohl inhaltlich als auch sprachlich angesprochen werden. Neben der Verwendung einer gendergerechten Sprache wird deshalb auch bei der Wahl der Themen und Beispiele auf Genderaspekte geachtet. Des Weiteren wird ein ausgewogenes Verhältnis von kompetitiven und kooperativen Aktivitäten der Lernenden angestrebt.

3.5 Lebensweltbezug

Das Projekt iLearnIT.ch versucht die behandelten Themen konsequent an Beispielen aus der Lebenswelt des Zielpublikums aufzuzeigen. Dieses Gestaltungsprinzip lässt sich sowohl mit dem *Sinnkriterium* [Sch93] als auch mit der Forderung nach *Gendergerechtigkeit* begründen und steigert auch schlicht die Attraktivität der Website für Kinder und Jugendliche. Schliesslich kann mit dem Lebensweltbezug Erwachsenen die Bedeutung von Informatikwissen für die Allgemeinbildung explizit oder implizit illustriert werden. Zusammen mit der Stufengerechtigkeit (siehe 3.3) kann dies konkret bedeuten, dass für die drei Schwierigkeitsstufen das gleiche Prinzip an drei verschiedenen Beispielen gezeigt wird.

3.6 Entdeckendes Lernen

Es gibt mehrere Gründe, warum entdeckendes Lernen bei iLearnIT.ch eine grosse Rolle spielt. Einerseits bietet sich dieser Ansatz bei einem Online-Lehrmittel an, das von Kindern und Jugendlichen zumeist selbständig und freiwillig genutzt wird. Bereits der erste Besuch der Website ist vermutlich oft eine Entdeckung. Dieses spielerische Interesse soll mit den Angeboten gefördert und genutzt werden. Gleichzeitig gehört das selbständige Erschliessen neuen Wissens zum Berufsalltag von Informatikerinnen und Informatikern [HNR06]. Insofern wird hier eine notwendige Berufskompetenz geübt bzw. geprüft. Konsequenterweise verzichtet iLearnIT.ch auf Tests, Prüfungen oder die Möglichkeit, ein iLearnIT.ch-Diplom zu erwerben. An gewissen Orten werden jedoch Selbsttests eingesetzt, um die Überprüfung des eigenen Verständnisses mangels Lehrperson zu ermöglichen.

3.7 computer science unplugged

Nach Vorbild des Neuseeländischen Projekts *computer science unplugged* [BBK07] wird bei iLearnIT.ch jedes Konzept auch mit einer Aktivität ohne jegliche Elektronik erfahrbar gemacht. Einerseits wird damit eine gängige Erwartungshaltung gebrochen, Informatik müsse immer mit Computern zu tun haben. Andererseits garantieren solche Aktivitäten sowohl die Betonung des Konzeptwissens gegenüber dem Produktwissens und machen Abstraktes konkret.

3.8 Mehrsprachigkeit

Die Schweiz besitzt vier offizielle Landessprachen. Das Projekt iLearnIT.ch wurde deswegen von Anfang an auf Mehrsprachigkeit ausgelegt. Derzeit sind Module auf deutsch und in französischer Übersetzung verfügbar. Mehrsprachigkeit ist jedoch nicht nur eine Frage der Übersetzung, hinter einer Sprache steckt auch eine Kultur und – für Bildungsprojekte relevant – pädagogische Überzeugungen. Obwohl bei der Erarbeitung auf deutsch darauf geachtet wird, dass Beispiele und Sprache möglichst sprachkultur-unabhängig gewählt werden (z.B. keine lokalen Sprichworte) bleibt trotzdem abzuwarten, wie sich die Akzeptanz von iLearnIT.ch in der französischsprachigen Schweiz entwickeln wird.

3.9 Scratch als Programmierumgebung

Obwohl Informatik nicht mit Programmieren gleichzusetzen ist, wird zur Vermittlung von informatischen Grundkonzepten auch eine Programmiersprache benötigt. Dabei gilt es zwischen vereinfachten Pseudosprachen, in realen Projekten genutzten Sprachen und speziell für Kinder und Jugendliche entwickelten Programmiersprachen auszuwählen.

Im Projekt iLearnIT.ch wird die vom MIT entwickelte, auf Smalltalk / Squeak aufbauende, für verschiedene Betriebssysteme frei verfügbare Programmierumgebung *Scratch* [MBK04] verwendet. Sie ermöglicht das einfache Erstellen von Programmen durch Zusammenfügen von vorgegebenen Programmelementen. Damit wird der Fokus auf die Konzepte und nicht die Sprachgrammatik gelegt. Im Vergleich zu älteren Programmierumgebungen für Bildungszwecke zeichnet sich Scratch durch grosse Multimedialität aus. Das Einbinden eigener Bilder und Töne erhöht die Attraktivität der Sprache für Kinder und Jugendliche und ermöglicht einen nicht-formalen Einstieg ins Programmieren, z.B. mittels story-telling, was unter Umständen das Interesse von Mädchen wecken kann [KP07]. Die Programmierumgebung Scratch eignet sich auch sehr gut für entdeckendes Lernen, passt also auch diesbezüglich zu den Prinzipien von iLearnIT.ch.



Abbildung 1: Die Roboter bit, byte und nibble

Mit Ausnahme des Intros (s. u. und Abbildung 1) wird Scratch bei iLearnIT.ch nicht nur eingesetzt, wenn Kinder und Jugendliche selbst etwas programmieren sollen, sondern mittels Java-Player auch bei allen Animationen und Beispielen. Der Verzicht auf die Verwendung von Adobe Flash oder eines anderen Multimediaautorentools erhöht zwar den Entwicklungsaufwand, senkt aber die Adaptationsschwelle der entwickelten Programme. Insbesondere soll damit die Gestaltbarkeit der Informatik praktisch gezeigt werden: Die Lernenden können alle Beispiele im Quellcode anschauen und weiter entwickeln.

4 Umsetzung

4.1 Rahmengeschichte

Um die geplanten zwölf Module in ein Ganzes einzubetten, wurde für iLearnIT.ch eine Rahmengeschichte erfunden: Die drei Roboter *bit*, *byte* und *nibble* (siehe Abbildung 1) sind auf der Erde gelandet und brauchen nun die Hilfe der Menschen, um sich hier zurechtzufinden und auch wieder heimkehren zu können. Die grafische Gestaltung der drei Roboter sowie die Umsetzung des Intros wurden durch externe Grafiker vorgenommen, um Kindern und Jugendlichen, die mit professionell gestalteten Medienwelten aufwachsen, einen ansprechenden Einstieg zu bieten. Bilder der drei Roboter in verschiedenen Posen stehen zum freien Download und Verwendung in eigenen Scratch-Projekten zur Verfügung.

4.2 Gesamtstruktur

Die ersten Terminplanungen vor Beginn des Schweizer Informatikjahres sahen vor, im Schweizer Jahr der Informatik jeden Monat ein Modul zu veröffentlichen. Aus diesem Grund ist die Gesamtstruktur von iLearnIT.ch auf zwölf thematische Module ausgelegt, je in zwei Sprachen (deutsch und französisch) und in drei Schwierigkeitsstufen (leicht, mittel, schwer). Dies ergibt insgesamt 72 Lerneinheiten (siehe Abbildung 2). Derzeit (Stand Mai 2009) sind drei Module auf Deutsch und ein Modul auf Französisch publiziert, weitere sind in Erarbeitung und Übersetzung.

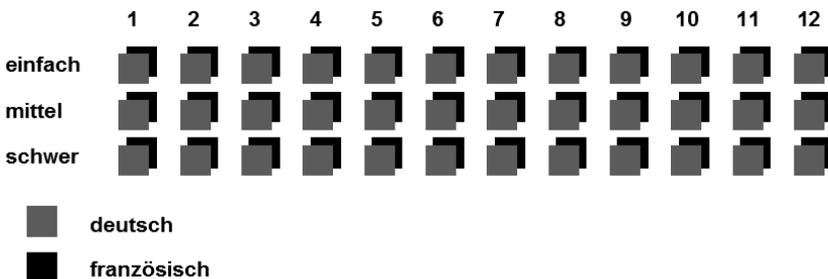


Abbildung 2: Geplante Gesamtstruktur von iLearnIT.ch

4.3 Struktur der Lerneinheiten

Jede der 72 geplanten Lerneinheiten von iLearnIT.ch soll gleich strukturiert sein. Zu diesem Zweck wurden die in Tabelle 2 abgebildeten Bereiche definiert. Sie sollen einerseits für die Nutzenden durch die Wiedererkennbarkeit die Navigation erleichtern. Andererseits sollen diese Bereiche sicherstellen, dass die oben beschriebenen Prinzipien von iLearnIT.ch auch zur Anwendung kommen und jede Lerneinheit die dafür notwendigen Elemente enthält.

Der zu Beginn des Projektes betriebene, relative grosse Aufwand zur Definition dieser Struktur sollte sich mit der Entwicklung weiterer Module wieder auszahlen, da nun vordefinierte „Gefässe“ gefüllt werden können. Dies wird besonders relevant, wenn Module durch externe Partner entwickelt werden sollen (s. u.).

Symbol	Bezeichnung	Erklärung
	Intro	Geschichte aus der Lebenswelt der Zielstufe
	Lösung suchen	Rätsel zum Experimentieren
	Tipps zur Lösung	Erklärungen und Hilfestellungen zur Lösung des Rätsels
	Selber programmieren	Hinweise für eigene Programme mit Scratch
	Ohne Computer	Enaktive Beschäftigung ohne Computer
	Forum	Austausch von Fragen und Rückmeldungen
	Links	Verweise auf andere Angebote im Internet
	Informationen für Lehrpersonen	Didaktische Hinweise und Lektionsideen

Tabelle 2: Elemente der einzelnen Lerneinheiten von iLearnIT.ch

5 Herausforderungen

Bereits das Entwickeln eines traditionellen, qualitativ hochstehenden Lehrmittels ist relativ aufwändig. Bei der Entwicklung von iLearnIT.ch haben sich folgende zusätzliche Herausforderungen gestellt:

- **Fehlende Vorbilder und -erfahrungen:** Im Gegensatz zu anderen Fächern sind in der Informatik nicht zahlreiche ältere Lehrmittel verfügbar, die als gute oder schlechte Beispiele bei der Entwicklung des eigenen Lehrmittels herangezogen werden können.
- **Fehlender allgemein akzeptierter und erprobter Lehrplan:** Nicht nur ältere Lehrmittel fehlen, sondern auch ein allgemein akzeptierter und bewährter Lehrplan für Informatik ist in der Schweiz derzeit nicht verfügbar. Zu vermittelnde Kompetenzen eines Informatik-Lehrmittels müssen somit zuerst selbst definiert werden.
- **Multimedialität:** Die Entwicklung eines multimedialen Lehrmittels ist aufwändiger als diejenige eines Buches. Neben Bild und Text müssen auch Ton, Video und interaktive Elemente geplant und umgesetzt werden. Dies erhöht die inhaltliche Komplexität, senkt aber gleichzeitig aufgrund technischer Formate die Lebensdauer des Lehrmittels. Bei den eingesetzten Multimedia-Formaten gilt es die Balance zwischen technisch Möglichem und beim Zielpublikum zuhause und in der Schule Vorhandenem und in der Schule nicht Gesperrtem (!) zu finden.

Zusammen mit den zahlreichen einzuhaltenden Gestaltungsprinzipien haben diese unterschätzten Herausforderungen dazu geführt, dass sich der vorgesehene Terminplan nicht einhalten liess. So konnten im Schweizer Jahr der Informatik „nur“ zwei Module auf Deutsch und eines auf Französisch publiziert werden. Es ist dabei aber zu berücksichtigen, dass die Module trotz vordefinierter Struktur umfangreicher geworden sind als geplant. So umfasst die deutschsprachige Version des Moduls 2 zum Thema Algorithmen in den drei Schwierigkeitsstufen 28 Scratch-Programme, 8 eigene und ein fremdes Video sowie 123 HTML-Seiten und 6 PDF-Dateien. Erst bei der Entwicklung kam die Erkenntnis, dass sich das Thema nicht schlanker vermitteln lässt: Unter anderem eine Folge der fehlenden älteren Informatik-Lehrmittel.

6 Ausblick

iLearnIT.ch soll auch nach dem Ende des Schweizer Jahres der Informatik weiter entwickelt werden. Mit der Fertigstellung von drei Modulen hat sich nun auch genügend Erfahrung angesammelt, welche einerseits die interne Entwicklung weiterer Module beschleunigen sollte. Andererseits ist auch angedacht, gewisse Module extern entwickeln zu lassen, um das vorhandene Material und Know-how der Informatik-Didaktik-Community nutzen zu können.

Es ist nun genügend Material vorhanden, um die Nutzung und Akzeptanz von iLearnIT.ch systematisch zu untersuchen. Bisherige Rückmeldungen haben zwar die Entwicklung beeinflusst, aber sie wurden nicht systematisch ausgewertet. So lassen sich derzeit keine gesicherten Aussagen zu Nutzung und Akzeptanz von iLearnIT.ch machen. Die individuellen Rückmeldungen zeigen aber, dass iLearnIT.ch bereits vereinzelt im Informatikunterricht im deutschsprachigen Raum eingesetzt wird.

Literaturverzeichnis

- [BBW07] Bell, T. C., Bensemam, G., Witten, I. H.: Computer Science Unplugged: Capturing the interest of the uninterested. <http://beat.doebe.li/bibliothek/t08464.html>
- [BCM06] Biundo, S.; Claus, V.; Mayr, H. C.: Was ist Informatik? Unser Positionspapier. Deutsche Gesellschaft für Informatik, 2006
<http://www.gi-ev.de/fileadmin/redaktion/Download/was-ist-informatik-lang.pdf>
<http://beat.doebe.li/bibliothek/t07972.html>
- [De03] Denning, P.J.: Great Principles of computing. Communicaions of the ACM, 46(11):15-20, November 2003. <http://beat.doebe.li/bibliothek/t05077.html>
- [De07] Denning, P.J.: Computing is a Natural Science. In: Communications of the ACM, 50(7):13-18, Juli 2007. <http://beat.doebe.li/bibliothek/t07784.html>
- [Dö09] Döbeli Honegger, B.: Biblionetz: Begriffe: Informatik
<http://beat.doebe.li/bibliothek/w00458.html>
- [HNR06] Hartmann, W.; Näf, M.; Reichert, R.: Informatikunterricht planen und durchführen, Springer Verlag, 2006. <http://beat.doebe.li/bibliothek/b02600.html>
- [ICH05] Genossenschaft I-CH: Modulbaukasten Informatik –R3, 2005,
http://www.i-ch.ch/modulbaukasten/pdf/modulbaukasten_r3d.pdf (Übersicht)
http://www.i-ch.ch/modulbaukasten/r3_grundbildung_de.html (Details)
- [KP07] Kelleher, C., Pausch, R.: Using Storytelling to motivate programming. In: Communications of the ACM. Volume 50, Issue 7 (July 2007)
<http://beat.doebe.li/bibliothek/t07792.html>
- [MBK04] Maloney, J., Burd, L, Kafai, Y., et al.: Scratch – A Sneak Preview. In: Proceedings of the Second International Conference on Creating, Connecting and Collaborating through Computing. <http://beat.doebe.li/bibliothek/t07843.html>
<http://llk.media.mit.edu/papers/ScratchSneakPreview.pdf>
- [MPFS08] Medienpädagogischer Forschungsverbund Südwest: KIM-Studie 2008.
<http://beat.doebe.li/bibliothek/b03536.html>
<http://www.mpfs.de/fileadmin/KIM-pdf08/KIM08.pdf>
- [PKH07] Pohl, W.; Kranzendorf, K.; Hein, H.: Einstieg Informatik, Aktivitäten und Erfahrungen. In: (Schubert, S. Hrsg.): Didaktik der Informatik in Theorie und Praxis, Proceedings der 12. GI-Fachtagung Informatik und Schule, GI-Edition- Lecture Notes in Informatics (LNI), P-112, 2007. <http://beat.doebe.li/bibliothek/t07902.html>
- [Ry04] Ryter, A. et al.: Kann jeder auch ein Mädchen sein? Leitfaden zur Erarbeitung von geschlechtergerechten Lehrmitteln, Lehrmittelverlag des Kantons Zürich, 2004.
<http://beat.doebe.li/bibliothek/b02285.html>
- [Sch93] Schwill, A.: Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik, 25(1):20-31, 1993. <http://beat.doebe.li/bibliothek/t01452.html>
- [SG08] Schleier, J.; Golliez, A.: Was ist Informatik? Broschüre im Rahmen der informatica08.ch <http://beat.doebe.li/bibliothek/t09275.html>
http://extra.informatica08.ch/Dokumente/informatica08/Brosch%C3%BCre_Was_ist_Informatic_de.pdf
- [Zeh08] Zehnder, C. A.: Erosion der Informatikausbildung, Vielfältige Ursachen, schwerwiegende Konsequenzen. In: Neue Zürcher Zeitung, 11.01.2008, <http://beat.doebe.li/bibliothek/t08215.html>,
http://www.nzz.ch/nachrichten/medien/erosion_der_informatikausbildung_1.649088.html

Informationskompetenz: ein Thema für den Informatikunterricht

Diana Jurjević, Nando Stöcklin, Werner Hartmann

Zentrum für Bildungsinformatik
Pädagogische Hochschule Bern
Muesmattstraße 29
CH-3012 Bern
diana.jurjevic@phbern.ch
nando.stoecklin@phbern.ch
werner.hartmann@phbern.ch

Abstract: Das Angebot an Online-Informationen nimmt weiter rasant zu. Die neuen Informationsmedien, speziell die Dienste im Web 2.0, erfordern neue Suchstrategien und neue Beurteilungsmethoden. Die Informationskompetenz heutiger Schülerinnen und Schüler lässt im Unterschied zu den Fertigkeiten und dem Gebrauch moderner Informations- und Kommunikationstechnologien aber zu wünschen übrig. Dem Informatikunterricht kommt hier eine zentrale Rolle in der Allgemeinbildung zu. Informationskompetenz bedingt auch ein Verständnis für die grundlegenden Konzepte von Informationsdiensten. Nach einem Überblick zu Informationsrecherche im Internet wird aufgezeigt, wie herkömmliche algorithmische Suchmaschinen und neuere Ansätze semantischer Suchdienste in Verbindung mit der geforderten Informationskompetenz im Informatikunterricht thematisiert und als authentische Lernumgebungen genutzt werden können.

1 Informationsrecherche im Internet, eine große Herausforderung

Der Umfang an Informationen im Internet ist in den letzten Jahren weiter rasant gewachsen. Besonders deutlich wird das am Web 2.0, dem Internet der Nutzer. Neben den etablierten Portalen wie Wikipedia, YouTube oder Facebook wächst ein gigantischer Ameisenhaufen von Microcontent, kleine und unorganisierte Informationseinheiten, wie ppt-Dokumente, Blogs oder Memos, jedoch ohne die persistenten Strukturen einer Ameisenkolonie. Diese Informationseinheiten werden nicht zentral verwaltet. Es gibt keine Stelle, welche die Inhalte zuerst sammelt, selektiert, hierarchisch anordnet und dann veröffentlicht, sondern die Inhalte liegen dezentral im Netz verteilt. Neben dem wachsenden Anteil an nutzergenerierten Inhalten im Web verändern sich auch die Formate der Inhalte. Die technischen Hürden, Fotos und Videos im Internet zu veröffentlichen, sind so gering geworden, dass diese Möglichkeit immer mehr genutzt wird. Damit hat eine Massenwanderung neuer Daten und Informationen in das Internet begonnen. 2008 betrug der Zuwachs neuer Videoclips allein auf YouTube circa 10.000 Stunden Videomaterial pro Tag. Für das Erzeugen von Bildern stehen heute weltweit mehr als

eine Milliarde Aufnahmegeräte zur Verfügung: Digitalkameras, Handys, Überwachungskameras oder Satellitenfotos sind nur einige Beispiele [Ga08].

Angesichts dieser Informationsflut hat das Selektieren von Informationen gegenüber dem Sammeln von Informationen deutlich an Bedeutung gewonnen. Gleichzeitig ist die Informationsbeschaffung auch anspruchsvoller geworden. Gerade um den Microcontent des Web 2.0 zu erschließen, müssen neue Suchstrategien entwickelt werden. Je nach Suchanfrage muss das passende Werkzeug gewählt werden: So gibt es für Blogs, Bilder und Videos spezialisierte Suchmaschinen. Die Herausforderung bei der Suche bleibt die Gleiche: Im Meer der Informationen sollen relevante Informationen von irrelevanten unterschieden werden. Bibliothekssysteme stützten sich auf Lösungsverfahren zum Auffinden von Informationen in großen Datenbeständen, welche die Informatik lange vor der Erfindung des Internets entwickelt hat. Heute verbreitet genutzte Internet-Suchmaschinen wie Google oder Yahoo sind Erben der Entdeckungen im Information Retrieval in den 1960er Jahre. Es handelt sich hier um *algorithmische* Suchmaschinen. Diesen Suchmaschinen gemeinsam ist, dass sie mittels Crawler die Inhalte des Webs durchsuchen, in einem Index erfassen und mit Hilfe statistischer Methoden zu einer Suchanfrage eine Auswahl und Rangierung der Inhalte vornehmen. Algorithmische Suchsysteme haben aber ihre Grenzen. Udi Manber, technischer Vizepräsident von Google, schildert, dass die heutigen Suchsysteme technologisch fast schon ausgereizt sind. Zur weiteren Suchmaschinenoptimierung sei heute vor allem ein besseres Verständnis der Nutzenden für die Funktionsweise von Suchmaschinen gefragt:

„It goes two ways: The content provider should think about how users will look for their content, and the user should think about what words people use to write about their content. Very often people make the mistake of using a search engine as if they are talking to another person. They use all sorts of words that a person will understand, but are not going to be in the content they are searching for. You should think about what you expect to see in the actual page and search for that. Having said that, we’re doing this, too. We will take your query and try to “understand” it and match it as best we can to the content we find on the Web.“ [Ma08]

Die Informationsbeschaffung wird zunehmend komplexer und mit dem Web 2.0 sind überdies neue, alternative Suchdienste entstanden, deren Funktionsweise sich deutlich abhebt von den bisher gängigen Suchmaschinen. Diese Suchdienste versuchen die bisherigen algorithmenorientierten Verfahren durch den Einbezug *semantischer* Technologien zu verbessern. So bieten zum Beispiel Social-Bookmarking-Dienste die Möglichkeit, auf den von den Benutzenden generierten Tags zu suchen, also in Inhalten, die durch die Nutzenden selbst verschlagwortet wurden. Diese Dienste können aber auch verwendet werden, um Inhalte geringer Qualität auszusortieren oder Gruppen von Nutzenden mit ähnlichen Interessen zu bilden. Eine gute Übersicht zum Status quo und den Entwicklungstendenzen bei Suchmaschinen findet sich in [LM08].

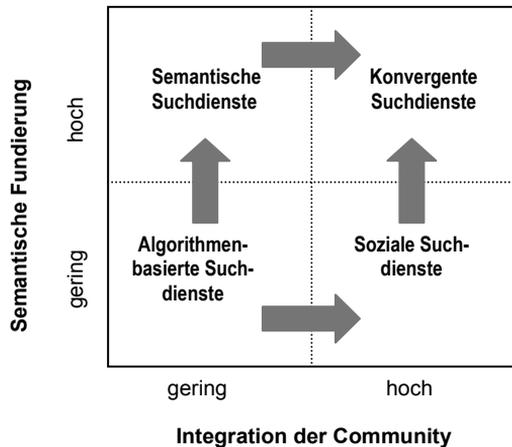


Abbildung 1: Unterschiedliche Typen von Suchdiensten, aus [LM08] S. 2

Für den Informatikunterricht bietet das Thema Suchsysteme ideale Voraussetzungen für eine authentische Lernumgebung, welche an den Alltagsproblemen der Schülerinnen und Schüler ansetzt, damit den Bezug zu ihrer Lebenswelt sicherstellt und Lernen hierdurch wahrscheinlicher macht. Der Informatikunterricht ermöglicht den Schülerinnen und Schülern, einen Blick hinter die Kulissen der kommerziellen Suchmaschinen zu werfen, die sie täglich nutzen. Grundlegende Konzepte von Informationsdiensten können aufgezeigt werden: Ausbeute und Präzision bei einer Suche, Indexierung als ein Schlüsselement in der maschinellen Durchforstung großer Datenmengen oder Boolesche Operatoren bei Datenbankabfragen sind nur einige Beispiele.

Das Wissen über die Funktionsweise von Suchmaschinen kann den Umgang mit Informationen im Internet verbessern. Dieter Baacke [Ba97] beschreibt in seinem vierdimensionalen Medienkompetenzmodell Medienkunde als eine Dimension von Medienkompetenz. Dem Informatikunterricht kommt bei der Internetrecherche eine Schlüsselrolle zu und hat das Potenzial, die Medienkompetenz der Schülerinnen und Schüler zu steigern.

2 Informationsdienste im Wandel

Die klassischen Internet-Informationendienste lehnen sich stark an Paradigmen aus der Bibliothekswelt an. Themenkataloge basieren auf der Einordnung von Informationen in Kategorien, algorithmische Suchmaschinen wie Google auf der Erschließung von Informationen anhand eines maschinengenerierten Stichwortindex. Die Forschungsdisziplin Information Retrieval untersucht und evaluiert Online-Suchdienste nach definierten Kriterien. Die klassischen Retrievalmaße stammen aus der Zeit vor dem großen Internetdurchbruch in den 1990er Jahren und berücksichtigen die Veränderungen nicht, die mit den Entwicklungen im Internet einhergegangen sind. Im Vordergrund stehen statistische Maße, das Nutzungsverhalten der User wird praktisch vollständig ausgeblendet. So spielt das optimale Verhältnis von Ausbeute und Präzision eine zentrale Rolle. Auf eine Such-

anfrage soll ein Suchdienst möglichst alle zur Verfügung stehenden relevanten Informationen anzeigen (Ausbeute), aber auch genau nur diese und keine Irrelevanten (Präzision). In der Praxis sagen aber diese Maße nur wenig über die Qualität einer Suchmaschine aus. 73% aller User, die eine Suchmaschine in den USA nutzen, schauen sich nur die erste Seite der Trefferlisten an, in Europa sind es sogar 83%. Diese Zahlen ermittelten Jansen und Spink [JS05] anhand der Webprotokolle von neun Suchmaschinen. Über eine Milliarde Suchanfragen wurden im Zeitraum von sieben Jahren gesammelt und miteinander verglichen. Suchmaschinen sortieren relevante Dokumente bereits im Vorfeld aus. Durch die Beschränkung auf die erste Trefferseite fallen auch noch alle relevanten Dokumente weg, die erst weiter hinten in den Trefferlisten erscheinen. Für die Evaluation von Suchsystemen bedeutet das, dass es nicht ausreicht, wenn ein Suchsystem relevante Dokumente findet, diese müssen auch auf der ersten Seite angezeigt werden. Alle relevanten Dokumente, die nicht auf der ersten Ergebnisseite erscheinen, nimmt die große Mehrheit der Nutzenden gar nicht wahr.

Je nachdem, ob der menschliche Faktor in die Evaluation einbezogen oder ausgeklammert wird, können die Studienergebnisse unterschiedlich ausfallen. Um die Systeme nachhaltig zu verbessern, müssen sich die Evaluationen vermehrt an der Realität der Suchenden und ihren Interaktionen orientieren. Auf diese Weise könnten neue, wirksame und nachhaltige Forschungsergebnisse gefunden werden. Bereits 1995 hielt Saracevic fest:

“The issue and challenge for any and all IR evaluations are the broadening of approaches and getting out of the isolation and blind spots of single level, narrow evaluation. How can interaction be ignored in IR evaluation at any level?” [Sa95]

Suchende nehmen nicht nur durch ihr Nutzungsverhalten Einfluss auf die Retrievalmaße, sondern sie bauen auch an neuen Arten der Informationserschließung im Internet mit. Web-2.0-Dienste stützen sich stark auf das Tagging ab. Nutzergenerierte Tags haben auf den ersten Blick den Nachteil der unkontrollierten Verschlagwortung durch Laien, der Alptraum aller Bibliotheksexperten. Bei genauerem Hinsehen stellt man aber fest, dass diese vermeintlichen Nachteile leicht durch die folgenden beiden Vorteile kompensiert werden können:

1. Von Experten vergebene Schlagworte bilden in der Regel das Wissen dieses Experten ab. Schlagworte basieren auf einem kontrollierten Thesaurus und sind deshalb oft orthogonal zur Sprachwelt von Laien. Demgegenüber sprechen die Benutzenden von Web-2.0-Diensten dieselbe Sprache wie andere Nutzende.
2. Da bei Social-Tagging-Diensten die Informationen von vielen Personen auf unterschiedliche Art verschlagwortet werden, führen Suchanfragen dort eher zum Erfolg als bei Verschlagwortung mit wenigen, in der Regel sehr spezifischen Begriffen durch Expertinnen.

Austauschplattformen wie Flickr, YouTube oder der Social-Bookmarking-Dienst Delicious sind damit eine Mischform zwischen von Experten erstellten Themenkatalogen und algorithmischen Suchmaschinen. Auf der einen Seite erschließen diese semantischen

oder sozialen Suchdienste weniger Informationen als die großen algorithmischen Suchmaschinen, die mit ihren Crawlern das Internet möglichst in der ganzen Breite und Tiefe absuchen. Rein statistisch gesehen nimmt damit die Ausbeute ab. Dieser Nachteil fällt aber kaum ins Gewicht, da die meisten User sowieso nur die ersten Treffer eines Suchdienstes inspizieren. Auf der anderen Seite bieten Suchdienste wie Delicious gewissermaßen von anderen Usern handverlesene Informationen an. Die Community dient als Filter für die Inhalte. Damit erhöht sich die Präzision bei der Suche.

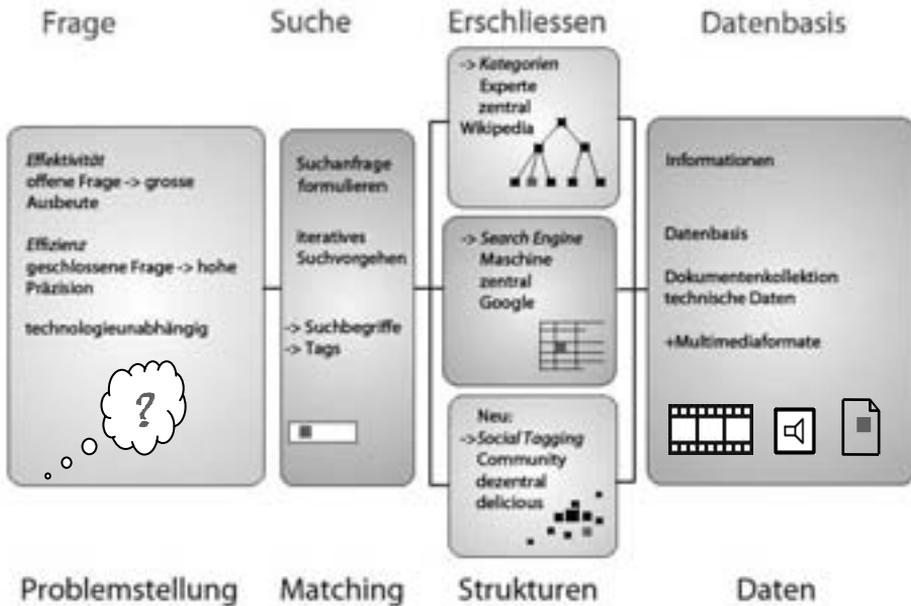


Abbildung 2: Prozessablauf der Informationssuche

Semantische und soziale Suchdienste sowie Mischformen zwischen diesen Diensten und herkömmlichen algorithmischen Suchdiensten sind in naher Zukunft aus der Landschaft der Informationsdienste nicht mehr wegzudenken. Information Retrieval ist nur ein Beispiel, wo informatische Konzepte und soziale Faktoren ein neues Ganzes bilden. Diese Entwicklung schlägt sich generell in der wachsenden Bedeutung der Web Science nieder, einem Wissenschaftszweig, der sich immer mehr neben der Computer Science etabliert und künftig im Informatikunterricht einfließen sollte.

Bislang wurden in der Informatik die Technologien mehr oder weniger getrennt von ihren Nutzerinnen und Nutzern behandelt. Der Bereich Human Computer Interaction fristet gerade an sehr technisch orientierten Hochschulen oft ein Schattendasein. Mit dem Web 2.0 lässt sich aber das Verhalten der Online-Nutzenden bei der Entwicklung von Informatiksystemen nicht mehr ausblenden. Ben Shneiderman fordert deshalb, das Forschungsfeld der Informatik um soziale Einflussfaktoren zu erweitern. Entwicklungen von Web-Technologien müssen Nutzerbedürfnisse berücksichtigen und deren Bedarf analysieren, bis hin zur Frage: Was wollen Personen und Communities vom Web

[Sh07]? Tim Berners Lee, der Erfinder des WWW, geht noch einen Schritt weiter. Er und weitere Wissenschaftler fordern, der Forschung über das WWW mehr Gewicht zu verleihen, indem neben der Informatik (Computer Science) die neue Disziplin „Web Science“ geschaffen wird [HS08]. Der tabellarische Vergleich der Disziplin „Web Science“ mit der bedeutend älteren Disziplin „Computer Science“ manifestiert den Trend im Internet: nicht mehr die Daten und Programme stehen im Vordergrund, sondern die Menschen, die sich für diese Daten und Programme interessieren.

Computer Science	Web Science
Topics	
Computer Networks	Social networks
Packet Switching	Voice over IP, music sharing
Information	Relationships
Programming languages	Wikis, blogs, tagging
Databases, operating systems, compilers	E-commerce, e-learning, e-government, medical informatics, financial analysis
3D graphics, rendering algorithms, computational geometry, object modelling	Creating and sharing video, animation, music, photos, maps

Abbildung 3: Computer Science vs. Web Science [Sh07]

3 Informationskompetenz im Informatikunterricht

Die Entwicklungen im Bereich der Suchdienste stellen die Nutzenden vor eine große Herausforderung. Die effiziente und effektive Informationsbeschaffung ist keineswegs trivial, auch wenn das viele Suchende glauben. In einer Studie des US-amerikanischen Pew-Forschungszentrums zeigt Deborah Fallows [Fa05] auf, dass Studierende reihenweise ihre Fähigkeiten bei der Suche überschätzen. Bei erfolglosen Suchprozessen neigen viele Nutzende zur Annahme, es gäbe die gesuchte Information im Internet nicht oder sie sei nicht auffindbar. Dieser Rückschluss greift oftmals zu kurz. Wahrscheinlicher ist, dass man die falschen Suchbegriffe gewählt, am falschen Ort gesucht, nicht die richtige Anzahl Suchbegriffe eingegeben oder den richtigen Treffer übersehen hat. Solche Fehleinschätzungen werden auf Paralyse zurückgeführt: Eine große Auswahlmöglichkeit verleitet gerade dazu, falsche Entscheidungen zu treffen, weil dann die Gefahr besteht, zu einfacheren Lösungs- und Erklärungsstrategien zu greifen oder die Entscheidung ganz aufzuschieben. Die Informationsflut im Internet ist ein Paradebeispiel dafür. Barry Schwartz (2004) beschreibt das Phänomen ausführlich in *The Tyranny of choice* im *Scientific American*.

Suchende überschätzen aber nicht nur sich selbst, sondern auch die Qualität ihrer Suchergebnisse. Das Vertrauen in die Richtigkeit der Informationen im Internet ähnelt dem Vertrauen in die etablierten TV- und Printmedien. Diese Erkenntnis ist nicht neu und ein klassisches Thema in der Medienpädagogik. Informationskompetenz ist eine Schlüsselqualifikation und umfasst nach der American Library Association die Fähigkeit, Informationsbedürfnisse zu erkennen, die benötigten Informationen gezielt zu lokalisieren und evaluieren und die gefundenen Informationen für die eigenen Zwecke nutzbar zu

machen und zu speichern. Für eine ausführliche Beschreibung sei etwa auf die Standards und Indikatoren zur Informationskompetenz von Studierenden von 1998 verwiesen [Am98].

Die kompetente Nutzung moderner Informationsdienste ist ein zunehmend komplexeres Unterfangen und setzt ein Verständnis für die Grundprinzipien der Suchdienste voraus. Genau so wie eine große Bibliothek ohne ein Verständnis für Katalogisierung, Oberbegriffe und Schlagworte nicht effizient genutzt werden kann, lassen sich Suchmaschinen nicht ohne dieses Hintergrundwissen kompetent nutzen. An dieser Stelle ist die Schule gefordert, die Grundlagen für die kompetente Nutzung dieser Dienste zu vermitteln. Genau hier kommt nun dem Informatikunterricht eine wichtige Rolle zu. Bereits 2004 hat die Gesellschaft für Informatik e.V. (GI) im Memorandum „Digitale Spaltung verhindern – Schulinformatik stärken!“ gefordert [Ge04]:

„Wenn die Informatik in der Schule stärker verankert wäre, würde dies die allgemeine IT-Kompetenz und somit letztlich die Effizienz zahlreicher Arbeitsprozesse erheblich erhöhen [...]. Um die allgemeine ICT-Kompetenz zu erhöhen, reicht es nicht, dass nur das Orientierungswissen gefördert wird, also nur Grundlagen wie Modellierung, Algorithmen und Datenstrukturen oder Berechnungsmodelle vermittelt werden. Ein Schulfach Informatik muss auch unmittelbar dazu beitragen, ICT-Werkzeuge effizient im Schulalltag und später im Arbeitsprozess zu nutzen. Es sollte deshalb neben dem Orientierungswissen die Anwenderkompetenz fördern.“

Der Aufbau und die gezielte Nutzung von Informationsdiensten muss ein integraler Bestandteil des Informatikunterrichtes sein. Es reicht aber nicht, nur die informatischen Konzepte von Suchsystemen zu lehren. Um den Unterricht zeitgemäss zu gestalten, müssen Faktoren wie Nutzungsverhalten und soziale Einflussfaktoren miteinbezogen werden. Am Beispiel zweier Lernumgebungen (Soekia und Tagidex) zeigen wir auf, wie sich im Informatikunterricht das Thema Informationsrecherche und Informationskompetenz sowohl aus informatischer Perspektive als auch für alle anderen Fächer gewinnbringend vermitteln lässt. Das Thema passt in verschiedene Inhaltsbereiche der Bildungsstandards Informatik, zum Beispiel Informatiksysteme, Strukturierungsprinzipien für Dokumente erkennen und anwenden und Informatik, Mensch und Gesellschaft [Ar07].

4 Soekia – ein Blick in die Blackbox algorithmischer Suchmaschinen

In der Medienpädagogik wurden zahlreiche Methoden entwickelt, die erfolgreich zur Annäherung der Schülerinnen und Schüler an ein Medium führen. So haben sich für das Medium Film Methoden wie die Filmanalyse oder das eigenständige Erstellen von Videoproduktionen vielfach bewährt, um die Verbindung zwischen Medium und Information bewusst zu machen. Diesem pädagogischen Ansatz folgt auch Soekia [DH03].

Die Arbeitsweise von algorithmischen Suchmaschinen bleibt den Nutzenden weitgehend verborgen: Das Erfassen von Webseiten durch Webroboter (Crawling / Spidering), das Erstellen einer effizienten Datenstruktur für die Suche (Indexierung), das Finden passen-

Die Lernumgebung wird auf verschiedenen Schulstufen ab der Sekundarstufe bis hin zur Fachhochschule eingesetzt. Eine typische Unterrichtssequenz erstreckt sich über 4-6 Lektionen und als Unterrichtsmethode bietet sich Entdeckendes Lernen besonders an. In einer kurzen Einleitung wird erläutert, wie der Crawler einer Suchmaschine Webseiten erfasst. Diese Komponente von Suchmaschinen ist in Soekia nicht enthalten. Anschließend erstellen die Schülerinnen und Schüler eine kleine Dokumentensammlung mit einigen wenigen kurzen Webseiten. Es hat sich bewährt, selbst mit einem einfachen Texteditor ein paar Webseiten zu einem bestimmten Thema zu erstellen. Lädt man existierende Webseiten herunter, verliert man schnell den Überblick. Selbst erstellte Seiten lassen sich leichter modifizieren, z. B. einzelne Worte mehrfach aufführen und dann den Einfluss der Häufigkeit des Vorkommens eines Suchbegriffes in einem Dokument auf die Rangierung beobachten.

Durch Einbezug verschiedener Wortformen des gleichen Wortstammes in den betrachteten Webseiten können die Auswirkungen der Wortstammnormalisierung auf den Index und damit zusammenhängend die Ausbeute einer Suche beobachtet werden. Durch die Normalisierung reduziert sich die Anzahl der Einträge im Index. Ein Suchbegriff findet damit auch andere Wortformen des Suchbegriffes, das heisst die Ausbeute der Suche nimmt zu und als unerwünschter, aber typischer Nebeneffekt die Präzision ab. In den begleitenden Unterrichtsmaterialien auf dem Bildungsserver SwissEduc ist eine ganze Reihe solcher möglicher Entdeckungen und Einsichten durch die Lernenden beschrieben.

Als interessante Herausforderung zum Abschluss einer Unterrichtssequenz mit Soekia hat sich ein Web-Spam-Wettbewerb erwiesen. Die Schülerinnen und Schüler erhalten den Auftrag, eine einfache Webseite zu einem vorgegebenen Thema (z.B. eine Autovermietung) zu erstellen. Ziel ist es, dass am Schluss die eigene Webseite bei einer typischen Suchanfrage aus dem vorgegebenen Themenkreis in der Rangliste möglichst weit oben erscheint. Neben der spielerischen Seite vermittelt ein solcher Wettbewerb auch tiefer gehende Einsichten zur Suchmaschinenoptimierung von Webseiten.

5 Tagidex

Soekia bietet die Grundlage für das Verständnis von volltextbasierten, algorithmischen Suchsystemen. Aufgrund der großen Nutzung von Soekia in unterschiedlichen Lehrkontexten arbeiten wir an einer neuen Lernumgebung, welche den neuesten Webentwicklungen Rechnung trägt. So soll die erweiterte Version den Schülerinnen und Schülern die Unterschiede zwischen einer Volltextsuche und einer tagbasierten Suche veranschaulichen. Dazu wird eine Webapplikation Tagidex entwickelt, welche es erlaubt, auf einer vergleichbar überschaubaren Kollektion von echten Websites zum Themenbereich ‚Elektrizität‘ mit Tagging- und Indexsuchverfahren zu suchen. Ein Prototyp von Tagidex findet sich auf [HJ09]. Die Nutzenden können selbst wählen, mit welchem Retrievalsystem die Kollektion durchsucht werden soll. Bei der tagbasierten Suche werden die dazugehörigen Tags von Social-Tagging-Diensten wie Delicious bezogen. Beispielsweise kann derselbe Suchbegriff in die Volltextsuche und in die tagbasierte Suche eingegeben werden, die Ergebnislisten können miteinander verglichen und charakteristische Merk-

malsunterschiede herausgearbeitet werden. Auf welchem Platz steht eine von Nutzenden verschlagwortete Seite im Vergleich zur Indexsuche? Wurde der erste Treffer auf eine Anfrage mit der Indexsuche auch von Nutzenden verschlagwortet? Stimmen die Schlagwörter ungefähr mit den Erwartungen überein? Tagidex verdeutlicht die unterschiedlichen Funktionsweisen der Suchsysteme. Die Schülerinnen und Schüler sollen lernen, ihre Suchanfragen zu differenzieren. Durch das Ausprobieren verschiedener Suchbegriffe und -systeme können sie erfahren, welche Suchstrategien mehr oder weniger erfolgreich sind.



Abbildung 5: Screenshot einer Suchergebnisseite aus Tagidex

Tagidex ist als didaktisches Hilfsmittel konzipiert und nicht als Lernumgebung für das Selbststudium gedacht. Bevor im Unterricht Tagidex zum Einsatz gelangt, braucht es eine Einführung in die Grundlagen der Internetrecherche, der tag- und indexbasierten Suchsysteme. Die Funktionsweise von indexbasierten Suchsystemen kann mit Soekia vermittelt werden. Bei tagbasierten Diensten wie Delicious muss zuerst die Nutzung als Suchmaschine aufgezeigt werden. Erst anschliessend können mit Tagidex Vor- und Nachteile der verschiedenen Erschliessungsmethoden erkundet werden. Es ergeben sich Fragestellungen wie z.B.: Was sind mögliche Gründe für die unterschiedliche Rangierung von Treffern bei den beiden Suchsystemen? Welche Kriterien müssen bei der Auswahl von guten Suchbegriffen berücksichtigt werden? Die Lernenden erkennen so grundlegende Unterschiede: bei indexbasierten Suchmaschinen muss man Suchbegriffe wählen, die im gewünschten Dokument enthalten sind. Bei tagbasierter Erschliessung muss man sich mehr die Benutzenden vorstellen und welche Tags diese verwendet haben könnten. Aufgrund erster Erprobungen im Unterricht kann vermutet werden, dass sich die Benutzenden mit beiden Typen von Suchsystemen zu Recht finden. Wer schon über gute Recherchekompetenzen mit herkömmlichen Suchmaschinen verfügt, nutzt zudem auch Social-Bookmarking-Dienste effizienter.

6 Suchdienste nach dem Prinzip „Do it yourself“

Mit Soekia und Tagidex stehen für den Informatikunterricht Lernumgebungen zur Verfügung, mit denen sich Experimente rund um Suchdienste durchführen lassen, bei Soekia auch auf größeren Dokumentensammlungen. So können etwa zuerst verschiedene Websites zu einem Themenbereich heruntergeladen und anschließend mit Soekia erschlossen werden. Beispielsweise könnten Websites verschiedener Autovermietungsfirmen gesammelt werden. Schülerinnen und Schüler könnten eine eigene Website für eine Autovermietungsfirma erstellen und dann die Rangierung im Vergleich zu den heruntergeladenen Websites untersuchen. Damit kann ein immer wieder auftretendes Problem im Informatikunterricht zumindest teilweise entschärft werden: Reale Informatiksysteme sind in der Regel äußerst komplex und entziehen sich den Möglichkeiten der Schule. In der Schule können nur kleine Systeme gebaut werden. Mit Soekia und Tagidex lassen sich zumindest große und reale Datenbestände im Unterricht einbinden.

Auf fortgeschrittener Stufe im Informatikunterricht wäre es spannend, selbst einen Suchdienst zu bauen. Die einfache Indexierung bei einer Volltextsuche, die Selektion der Resultate zu einer Suchanfrage und die Rangierung der Resultate nach einfachen Relevanzkriterien sind im Rahmen eines größeren Projektes durchaus auch für die Schule zugänglich. Es besteht die Möglichkeit, nur einzelne dieser Aspekte selbst umzusetzen und für andere auf vorgegebene Open-Source-Lösungen zurückzugreifen. Soekia selbst stützt sich auf die Suchmaschine Lucene von Apache ab. Der Quellcode von Soekia ist freigegeben unter der GNU GPL und könnte als Grundlage für weitere Entwicklungen dienen.

Das Erfassen von Websites in größerem Umfang übersteigt die Mittel im Informatikunterricht. Aber auch hier bieten sich inzwischen interessante Lösungen in Form von Web Services an. So macht beispielsweise Yahoo! im Rahmen von BOSS seinen Index allgemein zugänglich. Auch die Schule kommt damit in den Genuss der Infrastruktur von Yahoo! Die BOSS API ist einfach anzuwenden, die Resultate werden als XML-Antworten zurückgeliefert und können anschließend weiter bearbeitet werden. Wir sind überzeugt, dass der Informatikunterricht ganz allgemein vermehrt die heute zur Verfügung stehenden Web Services (Wetterdienste, News-Dienste usw.) und den damit verbundenen Zugang zu realen Daten nutzen sollte.

7 Zusammenfassung und Ausblick

Suchdienste im Internet werden im Informatikunterricht oft stiefväterlich behandelt. Sie eignen sich aber aus zwei Gründen hervorragend als Thema im Informatikunterricht:

- 1) Der Umgang mit Suchdiensten ist keineswegs trivial, sondern wird immer komplexer. Nur wer die Konzepte hinter den Suchdiensten versteht, kann diese auch gezielt nutzen. Die Vermittlung dieses Grundlagenwissens ist eine Aufgabe, die nur der Informatikunterricht wahrnehmen kann. Mit der Vermittlung von Informationskompetenz bewegt sich der Informatikunterricht in Richtung Web Science und Medienkunde. Das trägt zur Positionierung des Informatikunterrichtes in der Schullandschaft bei.

- 2) Am Beispiel von Suchdiensten lassen sich viele informatische Konzepte aufzeigen: effiziente Datenstrukturen für die Speicherung und Abfrage von großen Datenmengen. Algorithmen zur Selektion und Rangierung. Einbindung von Web Services als Schnittstelle zu großen realen Informatiksystemen.

Alle Beteiligten profitieren in mehrer Hinsicht. Suchdienste bieten ein Thema welches die Schülerinnen und Schüler anspricht, weil es einen direkten Anknüpfungspunkt an ihre Lebenswelt bietet. Den Informatiklehrenden wird damit ein Anker für den Unterrichteinstieg in die Hand gegeben. Für Lehrerinnen und Lehrer anderer Schulfächer ist Recherche und Informationskompetenz im Internet ein tägliches Thema. Aber es mangelt ihnen am nötigen Grundlagenwissen um selbst das Thema Informationskompetenz angemessen zu vermitteln. Hier kommt dem Informatikunterricht eine wichtige Rolle zu. Informatisches Wissen zu Suchdiensten würde die allgemeine IT-Kompetenz erhöhen und zahlreiche Arbeitsprozesse im Schulumfeld und Berufsalltag erleichtern.

Literaturverzeichnis

- [Am98] American Library Association: Information Literacy Standards for Student Learning. Chicago, 1998
- [Ar07] Arbeitskreis "Bildungsstandards": Grundsätze und Standards für die Informatik in der Schule. Beilage zu LOG IN, 146(147), 2007
- [Ba97] Baacke, Dieter: Medienkompetenz. Tübingen, 1997
- [DH03] Dreier, M.; Hartmann, W.: www.swisseduc.ch/informatik/soekia/ (Stand 08.01.2009)
- [Fa05] Fallows, D.: Search Engine Users. Internet searchers are confident, satisfied and trusting – but they are also unaware and naïve. PEW Internet & American Life Project, Washington, 2005
- [Ga08] Ganz, J.F.: The Diverse and Exploding Digital Universe. An Updated Forecast of Worldwide Information Growth Through 2011. IDC White Paper.
<http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf> (Stand 08.01.2009)
- [Ge04] Gesellschaft für Informatik (Hrsg.): Digitale Spaltung verhindern – Schulinformatik stärken! Memorandum, Ulm, 2004
- [HS08] Hender, J.; Shadbolt, N.; Hall, W.; Berners-Lee, T.; Weitzner, D.: Web Science: An Interdisciplinary Approach to Understanding the Web. Communications of the ACM, 51 (7), 2008; S. 60-69
- [HJ09] Hielscher, M.; Jurjević, D.: www.r2d2.ch/tagidex/ (Stand 08.01.2009)
- [JS05] Jansen, B. J.; Spink, A.: How are we searching the World Wide Web? A comparison of nine large search engine transaction logs. Information Processing and Management, 42(1), 2005; S. 248-263
- [LM08] Lewandowski, D.; Maass C. (Hrsg.): Web-2.0-Dienste als Ergänzung zu algorithmischen Suchmaschinen. Berlin, 2008
- [Ma08] Manber, U.: 20 (Rare) Questions for Google Search Guru Udi Manber,
http://www.popularmechanics.com/blogs/technology_news/4259137.html, 16. April 2008 (Stand 08.01.2009)
- [Sa95] Saracevic, T.: Evaluation of evaluation in information retrieval. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Special issue of SIGIR Forum, 1995; S. 138-146
- [Sh07] Shneiderman, B.: Web science: A Provocative Invitation to Computer Science. Communications of the ACM, 50(6), 2007; S. 25-27

Die visuelle Programmiersprache Puck – Entwicklung, Erprobung, Reflexion

Lutz Kohl
Fakultät für Mathematik und Informatik
Friedrich-Schiller-Universität Jena
lutz.kohl@gmail.com

Abstract: In diesem Artikel werden wesentliche Erkenntnisse aus der Dissertation des Autors vorgestellt. Puck ist eine visuelle Programmiersprache mit der grundlegende Konzepte imperativer Programmierung erlernt werden können. Bei einer Reflexion zeigte sich, dass existierende Empfehlungen zur Entwicklung interaktiver Lernumgebungen bei der Entwicklung von Puck nachvollzogen werden konnten. 40 Lehrkräfte, die Puck im Unterricht erprobten und anschließend einen Fragebogen beantworteten, empfanden das Verringern der Syntaxfehler durch Puck als günstig für den Anfangsunterricht. Weitere Ergebnisse der Befragung werden vorgestellt.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21. - 24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Kreativer Einstieg in die Programmierung Alice im Informatik-Anfangsunterricht

Michael Dohmen, Johannes Magenheim
Didaktik der Informatik
Universität Paderborn
Fürstenallee 11
33100 Paderborn
dohmen@upb.de
jsm@upb.de

Dieter Engbring
Gesamtschule Paderborn-Elsen
Am Schlengerbusch 27
33106 Paderborn
didier@upb.de

Abstract: In dem hier vorliegenden Artikel werden erste Ergebnisse einer Unterrichtsreihe mit Alice vorgestellt, die in der Jahrgangsstufe 9 an der Gesamtschule Paderborn-Elsen durchgeführt und evaluiert wurde. Dabei haben wir einen Motivationsschub beobachtet. Der Umgang mit Alice allein trägt jedoch nicht dazu bei, dass Schüler grundlegende Kenntnisse über Objektorientierte Programmierung erwerben. Der deklarative Zugang mit vorgefertigten Objekten erleichtert den Einstieg aber sehr.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21.–24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Adam Ries und das Linienrechnen – ein historisches Thema für den Informatikunterricht

Michael Fothe

Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
Ernst-Abbe-Platz 2
07743 Jena
michael.fothe@uni-jena.de

Abstract: Im Jahr 2009 wird der 450. Todestag des Rechenmeisters Adam Ries begangen. Das Linienrechnen, die mittelalterliche Version des Abakusrechnens, lässt dem Rechnenden beträchtliche Freiräume. Im Informatikunterricht bietet die Methode interessante Möglichkeiten zum Gewinnen von (Vor-)Erfahrungen zu Algorithmen, Nichtdeterminismus, Syntax, Semantik und zur Mechanisierung von Rechenvorgängen.

Vergesst nur die alten Meister nicht,
darauf wird aufgebaut
(N. J. Lehmann, 1.7.1990)

1. Zielstellung

Der Rechenmeister Adam Ries (1492-1559) setzte das Linienrechnen als methodische Vorstufe zum Ziffernrechnen ein [De92]. Das Rechnen auf den Linien besitzt auch heute noch Potenzial für die *Schulmathematik* und sollte daher in der Lehrerbildung thematisiert werden [Mö04], [He08], [Fo09]. In diesem Beitrag wird das Linienrechnen nun aus der Perspektive der *Schulinformatik* betrachtet. Immerhin wird das Rechnen auf den Linien z. B. in den Museumsführern durch die Informatik-Ausstellung im Deutschen Museum München und durch das Heinz Nixdorf Museumsforum (HNF) Paderborn erwähnt [Ba04], [HNF00]. In einem Plädoyer, die Geschichte der Informatik und der Informationsverarbeitung stärker in den Informatikunterricht zu integrieren, wird Adam Ries bei den potenziellen Schlüsselstellen in der Geschichte der Informatik unter der Überschrift „Entstehung von Zahlen, Zahlensysteme und Rechenregeln“ aufgeführt [Th05]. Eine Analyse anhand der GI-Empfehlungen zu Bildungsstandards Informatik zeigt, dass beim Linienrechnen zu relevanten Themen der Schulinformatik (Vor-) Erfahrungen gewonnen werden können, auf die im Unterricht an geeigneter Stelle zurückgegriffen werden kann [GI08]. In der Tabelle 1 sind entsprechende informatische Kompetenzen für die Sekundarstufe I zusammengestellt. Die Tabelle ist sicher noch erweiterbar. In den nachfolgenden Abschnitten werden Möglichkeiten zum Kompetenzerwerb im Informatikunterricht genauer beschrieben. Die Kompetenzen, die in einem Kompetenzmodell zum Inhaltsbereich „Algorithmen“ angegeben sind, könnten teilweise anhand von Algorithmen zum Rechnen auf den Linien entwickelt werden [KF07]. Das fächerüber-

greifende Thema „Adam Ries und das Linienrechnen“ ist sicher für die Unterrichtsfächer Mathematik, Informatik und Geschichte in der Sekundarstufe I von Interesse.

Nachfolgende Abschnitte	Kompetenzen aus den GI-Empfehlungen
2 (Adam Ries und seine Rechenbücher)	Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10 kennen und beachten <u>grundlegende Aspekte des Urheberrechts</u>
3 (Ein Beispiel für Syntax und Semantik)	Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10 unterscheiden die Begriffe „Syntax“ und „Semantik“ und erläutern sie an Beispielen
4 (Algorithmen zum Linienrechnen)	Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 interpretieren Handlungsvorschriften korrekt und führen sie schrittweise aus Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10 überprüfen die <u>wesentlichen Eigenschaften von Algorithmen</u>
5 (Das EVA-Prinzip und das Linienrechnen)	Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 erläutern das Prinzip der Eingabe, Verarbeitung und Ausgabe von Daten (EVA-Prinzip) als grundlegendes Arbeitsprinzip von Informatiksystemen
6 (Modellieren und Programmieren)	Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 bewerten Informationsdarstellungen hinsichtlich ihrer Eignung Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7 untersuchen bereits implementierte Systeme Schülerinnen und Schüler der Jahrgangsstufen 8 bis 10 analysieren Sachverhalte und erarbeiten angemessene Modelle

Tabelle 1: Kompetenzen mit Bezug zum Thema „Adam Ries und das Linienrechnen“.

2. Adam Ries und seine Rechenbücher

Adam Ries wurde 1492 in Staffelstein (Franken) geboren. Im Jahr 1518 erschien in Erfurt sein erstes Rechenbuch. Linienrechnen, praktische Aufgaben aus dem Wirtschaftsleben, das Beherrzigen des didaktischen Prinzips vom Einfachen zum Schwierigen und das ausführliche Beschreiben von Lösungsverfahren, nicht jedoch deren Begründungen waren Kennzeichen dieses Rechenbuches. Adam Ries betrieb wahrscheinlich in Erfurt (wie auch später in Annaberg) eine Rechenschule. 1522 erschien in Erfurt das zweite Rechenbuch. Das Rechnen auf den Linien wurde darin nur noch kurz gefasst, im Mittelpunkt stand das Ziffernrechnen. Die Aufgabensammlung wurde erweitert. Das zweite Rechenbuch begründete den Ruhm von Adam Ries. Über 100 Auflagen sind nachweisbar. In Erfurt begann Adam Ries auch die Arbeiten an seiner „Coß“ (eine „Wort-Algebra“). Die Erfurter Jahre waren zweifellos die wissenschaftlich fruchtbarsten im Leben des Rechenmeisters. Adam Ries übersiedelte 1522/23 nach Annaberg und war als Bergbeamter tätig (Rezessschreiber, Gegenschreiber, Zehntner). Im Jahr 1550 erschien sein drittes Rechenbuch in Leipzig. Es gilt als die beste deutsche Arithmetik in der Mitte des 16. Jahrhunderts [De92]. Das Manuskript war schon seit den 1520er Jahren fertig, jedoch behinderten hohe Druckkosten die Veröffentlichung. Adam Ries starb 1559 in Annaberg. Als Mathematiker war er auf der Höhe seiner Zeit, erbrachte jedoch praktisch keine eigenen originären Beiträge. Seine überragenden Verdienste liegen in der weiten Verbreitung des Rechnens in allen Bildungsschichten des Volkes. Die Riesschen

Rechenbücher kamen erst im 18. Jahrhundert allmählich außer Gebrauch. Adam Ries publizierte auf Deutsch und leistete damit einen wichtigen Beitrag zur Normierung der deutschen Sprache. *Für das dritte Rechenbuch erhielt Adam Ries auf Antrag ein zeitlich begrenztes Privileg des Kaisers Karl V., um den damals üblichen Raubdrucken zu begegnen. Den Schülerinnen und Schülern kann an diesem Beispiel deutlich werden, dass das Urheberrecht und dessen Sicherung bereits vor Jahrhunderten bedeutsam waren - und zwar sowohl im Interesse von Autor und Verlag als auch zum Nutzen für die Allgemeinheit. Die Diskussionen können zum Ausgangspunkt für das Thematisieren des Schutzes kreativer Leistungen in der Gegenwart werden.* Wegen weiterer Informationen zu Adam Ries wird auf die Literatur verwiesen [De92], [Ro08], [We92].

3. Ein Beispiel für Syntax und Semantik

Das Linienrechnen erfolgt auf einem Rechenbrett, Rechentuch oder Rechentisch und mit Rechenpfennigen. Wir nehmen ein Rechenbrett mit vier Linien, vier Zwischenräumen und drei Feldern (siehe Abbildung 1). Möglich wären auch andere Anzahlen.

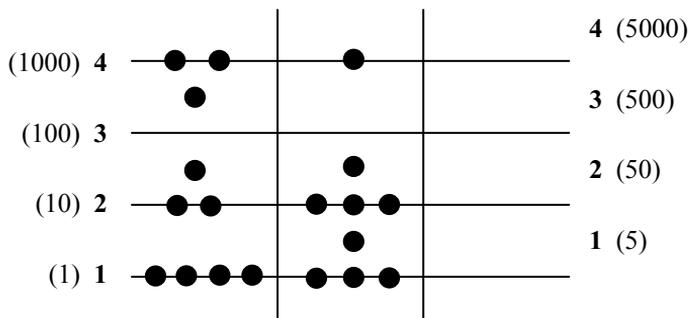


Abbildung 1: Die Linien und Zwischenräume des Rechenbrettes sind nummeriert. In Klammern ist der jeweilige Wert eines Rechenpfennigs angegeben. Im ersten Feld liegt die Zahl 2574, im zweiten Feld die Zahl 1088. Im Ergebnisfeld liegt kein Rechenpfennig, was der Zahl 0 entspricht.

Die beiden Zahlen, die addiert, multipliziert, subtrahiert oder dividiert werden sollen, werden in das erste und zweite Feld gelegt. Beim Legen einer Zahl gibt deren Einerstelle an, wie viele Rechenpfennige auf die Linie 1 und in den Zwischenraum 1 zu legen sind. Die Zehnerstelle der Zahl beschreibt das Belegen der Linie 2 und des Zwischenraums 2, die Hunderterstelle das Belegen der Linie 3 und des Zwischenraums 3 usw. Das Ergebnisfeld ist zu Beginn leer. Die Rechenpfennige werden nach speziellen Regeln verschoben, ersetzt, weggenommen oder hingelegt. Von grundlegender Bedeutung für das Rechnen auf den Linien sind die Ersetzungsregeln, die das Höherlegen (Elevieren) und Tieferlegen (Resolvieren) von Rechenpfennigen beschreiben (siehe Tabelle 2). Höher- oder Tieferlegen verändern die Summe der Werte aller Rechenpfennige in einem Feld nicht.

H1	Zwei Rechenpfennige, die in einem Zwischenraum liegen, können weggenommen werden. Dafür wird dann ein Rechenpfennig auf die nächsthöhere Linie gelegt.
H2	Fünf Rechenpfennige, die auf einer Linie liegen, können weggenommen werden. Dafür wird dann ein Rechenpfennig in den nächsthöheren Zwischenraum gelegt.
T1	Ein Rechenpfennig, der auf einer Linie liegt, kann weggenommen werden. Dafür werden dann zwei Rechenpfennige in den nächsttieferen Zwischenraum gelegt.
T2	Ein Rechenpfennig, der in einem Zwischenraum liegt, kann weggenommen werden. Dafür werden dann fünf Rechenpfennige auf die nächsttiefere Linie gelegt.
T3	Ein Rechenpfennig, der auf einer Linie liegt, kann weggenommen werden. Dafür werden dann ein Rechenpfennig in den nächsttieferen Zwischenraum und fünf Rechenpfennige auf die nächsttiefere Linie gelegt.

Tabelle 2: Die fünf Ersetzungsregeln, nach denen Rechenpfennige höher- oder tiefergelegt werden können. Die Ersetzungsregel T3 ist speziell für das Subtrahieren vorgesehen.

Die Ersetzungsregeln können als grafikorientierte Produktionsregeln einer formalen Sprache formuliert werden (siehe Abbildung 2).

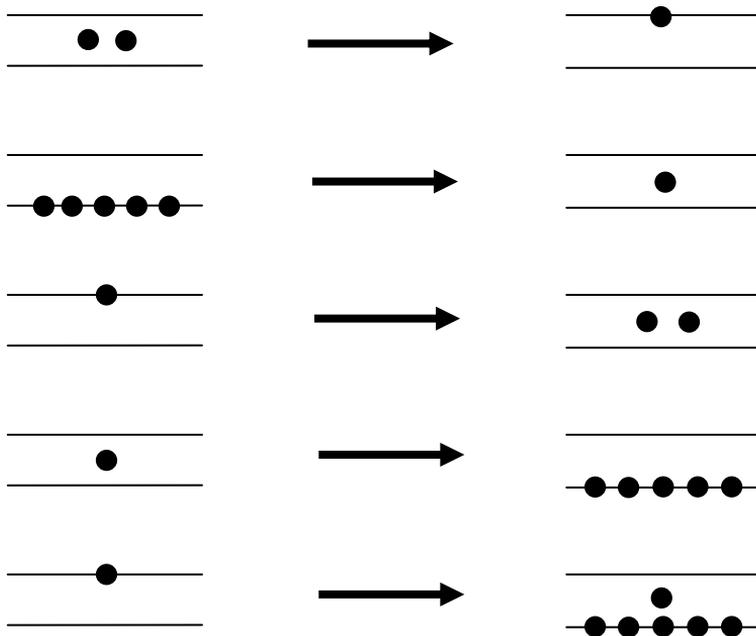


Abbildung 2: Formale Darstellung der Ersetzungsregeln (siehe Tabelle 2).

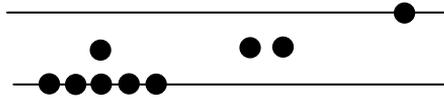


Abbildung 3: Beispiel für das Höherlegen von Rechenpfennigen. Die Ersetzungsregeln H2 und H1 werden nacheinander angewandt. Beim Höherlegen der fünf Rechenpfennige entsteht ein primärer Übertrag, der einen sekundären Übertrag zur Folge hat.

Das Rechnen ist beendet, wenn im Ergebnisfeld auf den Linien jeweils höchstens vier Rechenpfennige und in den Zwischenräumen jeweils höchstens ein Rechenpfennig liegt. Überzählige Rechenpfennige sind höherzulegen (siehe Abbildung 3). Das Höherlegen verändert, wie bereits dargelegt, den Gesamtwert der Rechenpfennige im Ergebnisfeld nicht. Es führt jedoch zu syntaktisch korrekten Ziffern, so dass das unmittelbare Ablesen des Resultates möglich wird (siehe Abbildung 4). Anhand des Linienrechnens können die Begriffe „Syntax“ („Wie ist etwas korrekt aufzuschreiben?“) und „Semantik“ („Was bedeutet das Aufgeschriebene?“) im Informatikunterricht thematisiert werden. Dabei werden grafische Objekte und nicht wie meistens Zeichen betrachtet.

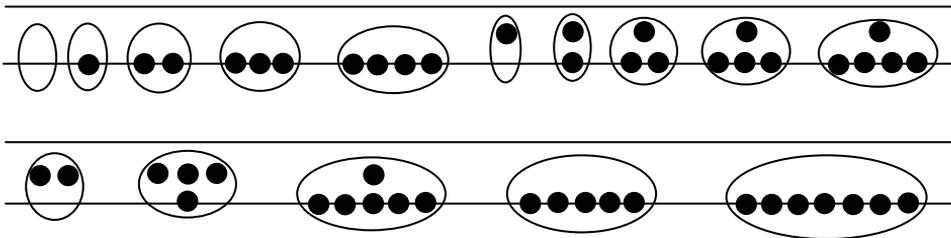


Abbildung 4: Die obere Zeile enthält die syntaktisch korrekten Ziffern. In der unteren Zeile liegen Syntaxfehler vor.

4. Algorithmen zum Linienrechnen

Operationen beim Linienrechnen sind Verdoppeln (Duplieren), Halbieren (Medieren), Addieren (Summieren), Multiplizieren, Subtrahieren und Dividieren [De92], [Fo09]. Die Rechenregeln sind arithmetisch begründet. Diese Tatsache ist beim Ausführen jedoch nebensächlich. Die Schülerinnen und Schüler sollen die Rechenregeln nach Möglichkeit selbst entdecken und auch begründen. Sie lernen mit dem Linienrechnen *konkrete Algorithmen* kennen und können diese ohne Computer zur Abarbeitung bringen. Sie können überprüfen, ob die Beschreibungen für das Rechnen auf den Linien die wesentlichen *Eigenschaften von Algorithmen* erfüllen (eindeutig, ausführbar, allgemein, endlich). Sie können erkennen, dass an gewissen Stellen mehrere Möglichkeiten der Fortsetzung bestehen, von denen sie nach Belieben eine auswählen können, dass jedoch beim Berechnen einer ganz bestimmten Aufgabe stets das gleiche Ergebnis herauskommt – vorausgesetzt natürlich, sie verrechnen sich nicht. Beim Linienrechnen kommt es häufig nicht auf die Reihenfolge von Handlungen an. Das kann geradezu als ein Kennzeichen dieser Methode angesehen werden. Anhand des Linienrechnens können die Schülerinnen und

Schüler somit Vorerfahrungen zum *Konzept des Nichtdeterminismus* gewinnen [CS06]. Ein Beispiel ist das Höherlegen im Ergebnisfeld, das ständig erfolgen kann oder zwischendurch immer mal wieder oder nur ganz zum Schluss; der Bedarf an eingesetzten Rechenpfennigen kann dabei sehr unterschiedlich sein. Ein anderes Beispiel ist das Multiplizieren, das in unterschiedlichen Varianten durchgeführt werden kann. Es gibt Varianten, bei denen das Beherrschen des Einmaleins nicht erforderlich ist, und es gibt Varianten, bei denen man das Einmaleins zumindest teilweise beherrschen muss. Das Verwenden des Einmaleins macht das Multiplizieren häufig schneller, weil vergleichsweise wenige Rechenpfennige zu bewegen sind. Das Rechnen auf den Linien lässt sich als *Musterverarbeitung* charakterisieren. Dabei wird das Linienrechnen aus der Perspektive der Informatik betrachtet und es wird eine Abstraktion vorgenommen, um das Wesentliche des Rechenverfahrens zu verdeutlichen [Fo04]. Diese Variante zum Multiplizieren soll am Beispiel 125×481 erläutert werden (siehe Abbildung 5). Das linke Rechenbrett stellt die Ausgangssituation dar. Im rechten Rechenbrett ist die Situation nach dem Bearbeiten eines der beiden Rechenpfennige von Linie 2 dargestellt. Dieser Rechenpfennig ist weiß hervorgehoben und wird dann weggenommen. Die Anordnung vom zweiten Feld wurde vollständig in das Ergebnisfeld gelegt – und zwar um eine Linie nach oben verschoben. Rechenpfennige im ersten Feld können in beliebiger Reihenfolge bearbeitet werden. Der Rechenpfennig im Zwischenraum 1 ist vor der Bearbeitung tieferzulegen.

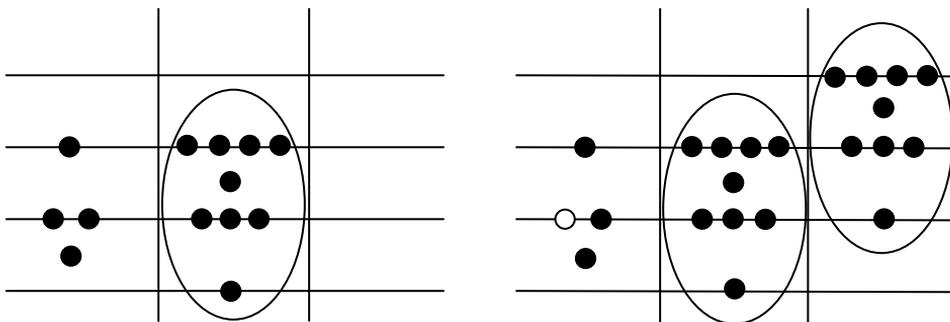


Abbildung 5: Berechnung von 125×481 .

Wenn beim Multiplizieren der gerade bearbeitete Rechenpfennig im ersten Feld weggenommen wurde (sozusagen ein Arbeitstakt abgeschlossen wurde), gilt

$$\text{ErstesFeld} \times \text{ZweitesFeld} + \text{ErgebnisFeld} = \text{Resultat.}$$

Aus dieser Gleichung folgt, dass jetzt die Belegungen des ersten und zweiten Feldes getauscht werden dürfen, ohne dass das Ergebnis falsch wird (siehe Abbildung 6). Man kann fragen, ob sich diese Eigenschaft zur Erlangung von Rechenvorteilen ausnutzen lässt. Im Übrigen muss das Tauschen nicht „physisch“ erfolgen; vielmehr kann das erste Feld (evtl. zeitweise) die Nummer 2 erhalten, das zweite Feld die Nummer 1.

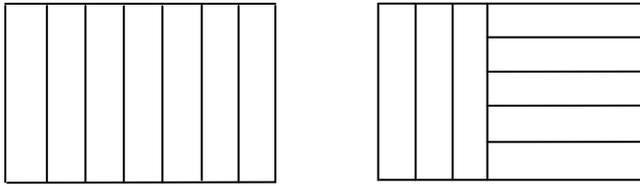


Abbildung 6: Links ist für das Beispiel 7×5 das übliche Zurückführen der Multiplikation auf die Addition visualisiert. Rechts ist die Situation dargestellt, dass die Belegungen des ersten und zweiten Feldes mitten in der Berechnung (einmal) getauscht werden. Es wird $3 \times 5 + 5 \times 4$ gerechnet.

Zu Beginn des Addierens, Multiplizierens, Subtrahierens oder Dividierens müssen keine syntaktisch korrekten Ziffern vorliegen. Bei Aufgaben wie $(36 + 81) \times 4$ ist es daher nicht erforderlich, nach der Berechnung von $36 + 81$ überzählige Rechenpfennige höherzulegen. Man kann „einfach“ weiterrechnen und mit 4 multiplizieren. Die *Robustheit* des Linienrechnens ist schon beträchtlich.

Nachteilig beim Linienrechnen gegenüber dem Ziffernrechnen ist, dass Zwischenergebnisse nicht von Natur aus dokumentiert werden¹. Im Mittelalter wurden Rechnungen auf den Linien, um Rechenfehler und Betrug weitgehend auszuschließen, mitunter von zwei Personen gleichzeitig ausgeführt. Dieses Vorgehen zum Erreichen der *Korrektheit* gab es auch in der Frühphase der Computertechnik, und zwar bei der Oprema, die 1954/55 im Zeisswerk Jena entwickelt wurde². Die Oprema war als Zwillingemaschine konzipiert, die aus zwei baugleichen Anlagen besteht. Alle Rechenschritte sollten redundant in beiden Anlagen ausgeführt werden. Nachdem sich aber gezeigt hatte, dass beide Anlagen zuverlässig arbeiten, wurde die ursprüngliche Idee fallengelassen und man hatte zwei Computer zur Verfügung.

Vorerfahrungen zur Problematik des *Zeitverhaltens von Algorithmen* können anhand des Linienrechnens gewonnen werden. Beim Linienrechnen kann man Umwege gehen. Umwege führen zwar zu mehr Aufwand, nicht jedoch zu Rechenfehlern. Ein Beispiel ist das Dividieren, bei dem Rechenpfennige im ersten Feld unnötig tief gelegt werden können; dafür sind dann später im Ergebnisfeld Rechenpfennige höherzulegen. Wie lässt sich der Aufwand bestimmen? Eine Möglichkeit ist das Ermitteln der Anzahl an Schritten, die auszuführen sind. Beim Dividieren auf den Linien können die folgenden Schritte betrachtet werden: 1. Tieferlegen im ersten Feld, 2. Wegnehmen im ersten Feld/Legen im Ergebnisfeld und 3. Höherlegen im Ergebnisfeld.

Ein gutes Resümee zu den Algorithmen gibt Menninger: „Ich empfehle dem Leser, unsere Aufgabe wirklich auf den Linien durchzuführen; er wird erstaunt sein über die Anschaulichkeit des Verfahrens, das ohne eigentliches Rechnen vor sich geht.“ [Me79]

¹ Beim Rechnen mit dem Taschenrechner ist das auch so.

² <http://idw-online.de/pages/de/news284008>; geprüft: 4. Mai 2009

5. Das EVA-Prinzip und das Linienrechnen

Den Schülerinnen und Schülern kann auch am Linienrechnen deutlich werden, dass das EVA-Prinzip ein wirklich grundlegendes Prinzip ist. Beim Rechnen auf den Linien werden indisch-arabische oder ursprüngliche römische Zahlen ein- und ausgegeben. Beispiele für solche römischen Zahlen³ sind MMDLXXIII und MLXXXVIII von Abbildung 1. Die Verarbeitung geschieht durch das Operieren mit Rechenpfennigen. Nur mithilfe eines Abakus oder mit Linienrechnen kann produktiv mit römischen Zahlen gerechnet werden. Das Rechnen auf den Linien kann als Vorstufe zu mechanischen Rechenmaschinen betrachtet werden. Wenn Schülerinnen und Schüler auf den Linien rechnen, simulieren sie in gewisser Weise eine mechanische Rechenmaschine. Die folgenden Erkenntnisse können von den Schülerinnen und Schülern gewonnen werden:

- Das Multiplizieren auf den Linien wird auf das Addieren, das Dividieren auf das Subtrahieren zurückgeführt. Das ist auch bei mechanischen Rechenmaschinen so.
- Bei mechanischen Rechenmaschinen gibt es in der Regel den Zehnerübertrag. Einen direkten Zehnerübertrag gibt es beim Rechnen auf den Linien nicht. Stattdessen liegen Fünfer- und Zweierüberträge vor.
- Bei mechanischen Rechenmaschinen werden Überträge „von unten nach oben“ – also ausgehend von der Einerstelle – bearbeitet. Beim Linienrechnen kann man das so machen, man muss es aber nicht.
- Primäre und sekundäre Überträge werden beim Linienrechnen gleichartig behandelt.

6. Modellieren und Programmieren

Bereits in der Sekundarstufe I kann das Modellieren von Rechenbrett und Rechenpfennig aus objektorientierter Perspektive erfolgen [Hu07]. Computerprogramme zum grafischen Darstellen von syntaktisch korrekten Ziffern, zum Legen einer Zahl auf die Linien und Zwischenräume und zum Höher- oder Tieferlegen von Rechenpfennigen können von den Schülerinnen und Schülern analysiert bzw. entworfen und implementiert werden.

Das Rechnen auf den Linien wurde vom Autor in Spielform beschrieben [Fo01]. Zu den Einpersonenspielen wurde ein Computerprogramm entwickelt. Der Spieler teilt dem Computer die Arbeitsschritte für das Addieren, Multiplizieren, Subtrahieren und Dividieren mit. Ein Beispiel für einen Arbeitsschritt, der beim Multiplizieren auszuführen ist, ist die Angabe, welcher Rechenpfennig im ersten Feld als nächstes bearbeitet werden soll. Der Computer führt den Arbeitsschritt aus und stellt das jeweilige (Zwischen-)Resultat grafisch dar. Das Computerprogramm lässt Regelverstöße durch den Spieler (den Rechnenden) nicht zu. Dadurch kann er sich auf das Umsetzen einer Spielstrategie (Re-

³ Bei den ursprünglichen römischen Zahlen gibt es keine verkürzenden Schreibweisen wie beispielsweise IV statt IIII. Diese bürgerten sich erst zu Beginn des 16. Jahrhunderts ein [BG91].

chenstrategie) konzentrieren. Wesentliches Ziel der Strategie ist das schnelle Gewinnen des Resultates.

Im Informatikunterricht der Sekundarstufe II können die Schülerinnen und Schüler das vorgegebene Computerprogramm analysieren, um die zugrunde liegenden Algorithmen zu verstehen. Anschließend können sie die Benutzungsfreundlichkeit des Programms erhöhen, die Bildschirmgestaltung verbessern oder den Leistungsumfang erweitern. Möglich wäre auch ein Umsetzen des vorgegebenen Programms in andere Programmiersprachen. Im Rahmen von Projektarbeiten könnten die vollständigen Regeln für das Linienrechnen mit dem Ziel der Visualisierung der Rechenvorgänge in Computerprogramme überführt werden⁴.

7. Resümee

Der Autor hielt anlässlich des 450. Todestages mehrere Vorträge zu Adam Ries und dem Linienrechnen. Im Wintersemester 2008/09 thematisierte er das Linienrechnen in einem Modul zur Fachdidaktik Informatik an der Universität Jena (Grundlage dafür war eine Entwurfsfassung dieses Beitrags). Das Interesse der Zuhörer, Näheres zu der historischen Art des Rechnens zu erfahren, war dabei stets vorhanden. Das Linienrechnen übt anscheinend eine gewisse Faszination aus. Diese Faszination kann Ausgangspunkt für das Behandeln des Linienrechnens im Informatikunterricht sein. Mögliche fachliche Anknüpfungspunkte und didaktisch-methodische Ideen sollte dieser Beitrag aufzeigen. Weitergehende Arbeiten sind möglich. So könnten Werkzeuge wie Tabellenkalkulationsprogramme auf ihre Verwendbarkeit hin überprüft werden.

Referenzen

- [Ba04] Bauer, F. L.: Informatik. Führer durch die Ausstellung. Deutsches Museum München 2004.
- [BG91] Bauer, F. L.; Goos, G.: Informatik 1. Eine einführende Übersicht. 4. Aufl. Springer Berlin, Heidelberg, New York 1991.
- [CS06] Claus, V.; Schwill, A.: Duden Informatik A-Z. 4. Aufl. Dudenverlag Mannheim 2006.
- [De92] Deschauer, S.: Das zweite Rechenbuch von Adam Ries. Friedr. Vieweg Braunschweig, Wiesbaden 1992.
- [Fo01] Fothe, M.: Rechnen auf den Linien. Vier Spiele mit dem Computer. LOG IN Heft 3/4 (2001), S. 48-53.
- [Fo04] Fothe, M.: Rechnen auf den Linien – eine historische Betrachtung aus der Sicht der modernen Informatik. In: Roloff, H.; Weidauer, M. (Hrsg.): Wege zu Adam Ries. Tagung zur Geschichte der Mathematik Erfurt 2002. Dr. Erwin Rauner Verlag Augsburg 2004, S. 87-91.
- [Fo09] Fothe, M.: Adam Ries und das Rechnen auf den Linien. Begleitmaterial für die Lehrerfortbildung. In: Jenaer Schriften zur Mathematik und Informatik 01/09 (http://www.minet.uni-jena.de/preprints/fothe_09/Fothe-Linienrechnen.pdf; geprüft: 4. Mai 2009).

⁴ <http://users.minet.uni-jena.de/~infotest/ries/applet/AdamRies-start.html>; geprüft: 4. Mai 2009

- [GI08] Gesellschaft für Informatik (Hrsg.): Grundsätze und Standards für die Informatik in der Schule. Bildungsstandards Informatik für die Sekundarstufe I. Beilage zu: LOG IN Heft 150/151 (2008).
- [He08] Hempel, T.: Adam Ries – Rechnung auf der linihen (<http://www.tinohempel.de/info/mathe/ries/ries.htm>; geprüft: 4. Mai 2009).
- [HNF00] Heinz Nixdorf Museumsforum: Museumsführer. Paderborn 2000.
- [Hu07] Hubwieser, P.: Didaktik der Informatik. 3. Aufl. Springer Berlin, Heidelberg, New York 2007.
- [KF07] Kohl, L.; Fothe, M.: Algorithmen aus einer anderen Perspektive. Ein Vorschlag für ein Kompetenzmodell zum Inhaltsbereich „Algorithmen“ der „Bildungsstandards Informatik“. LOG IN Heft 146/147 (2007), S. 20-22.
- [Me79] Menninger, K.: Zahlwort und Ziffer. Eine Kulturgeschichte der Zahl. 3. Aufl. Vandenhoeck/Ruprecht Göttingen 1979.
- [Mö04] Möller, R. D.: Das didaktische Potenzial des Linienrechnens heute. In: Roloff, H.; Weidauer, M. (Hrsg.): Wege zu Adam Ries. Tagung zur Geschichte der Mathematik Erfurt 2002. Dr. Erwin Rauner Verlag Augsburg 2004, S. 267-283.
- [Ro08] Rochhaus, P.: Adam Ries. Vater des modernen Rechnens. Sutton Verlag Erfurt 2008.
- [Th05] Thomas, M.: Vom Abacus bis Zuse. In: Friedrich, S. (Hrsg.): Unterrichtskonzepte für informatische Bildung. 11. GI-Fachtagung Informatik und Schule INFOS 2005. Lecture Notes in Informatics, Bonn 2005, S. 185-196.
- [We92] Weidauer, M.: Der Erfurter Rechenmeister Adam Ries. Erfurt 1992.

Computus

Der Weg eines Begriffs durch die Geschichte

Jürgen Müller

Berufsakademie Gera
Weg de Freundschaft 4A
D-07546 Gera
juergen.mueller@ba-gera.de

Abstract: Im Beitrag wird den Wurzeln von informationsverarbeitenden Systemen am Beispiel der Entwicklung des Begriffs *Computer* nachgegangen. Die Entwicklung des *computus* des Mittelalters, der die historische Quelle und Vorstufe für diesen Begriff bildet, wird über die Jahrhunderte bis in die Neuzeit verfolgt. Implementationen des *computus* werden dargestellt, die Vorläufer heutiger Computer sind.

1 Computer versus Calculator

Unzweifelhaft steht er im Mittelpunkt der Inhaltsbereiche der informatischen Bildung: der *Computer*. Als Lehrkräfte machen wir deutlich, wie er als zentraler Knoten in Informatiksystemen eingegebene Daten in Informationen verwandelt, dabei von Algorithmen gesteuert wird, die in speziellen Sprachen formuliert werden [GI08, S. 11]. Woher aber kommt dieser Begriff, der heute ohne groß nachzudenken in aller Munde ist, wo liegen seine historischen Wurzeln? Am 12. April 1963 soll dieses Wort in Deutschland in der ZEIT zum ersten Mal gedruckt worden sein. Im englischen Sprachraum taucht der Begriff deutlich früher auf, zuerst vermutlich im Jahr 1897 in einer englischen Ingenieurzeitschrift. Und so erläutern wir den Schülerinnen und Schülern meist: Computer kommt über die englische Sprache, in der *to compute* bekanntlich mit „zählen, rechnen“ übersetzt wird. Aber warum dann nicht *calculator*? *To calculate* ist ebenfalls „rechnen“, sogar im anspruchsvollen Sinn, denn Kurse der höheren Mathematik werden in Nordamerika als *calculus* bezeichnet. Das führt deshalb zu Recht heute zum *pocket calculator*, dem Taschenrechner. Was hebt nun aber den Computer von diesem Begriff ab? Was hieß dieses Wort früher, was wollten die Urväter der Informatik mit dieser Assoziation ansprechen? Ist dieser Begriff quasi vom Himmel heruntergefallen? In gewisser Weise schon, denn Sonne, Mond und Sterne spielen bei der Begriffsbildung eine wichtige Rolle! Es ist der *computus* des Mittelalters, der die historische Quelle und Vorstufe für diesen Begriff bildet. Der *computus* war die Wissenschaft von der Zeitrechnung, genauer der Vorausberechnung des Osterdatums.

Aus heutiger Sicht stellt sich der *computus* als eine Sammlung von komplexen Algorithmen zur Osterrechnung dar. Neben dem Studium der Heiligen Schrift gehörte der *computus* zu den Bildungsstandards bei der theologischen Ausbildung der Geistlichen im Mittelalter, war in gewisser Weise also eine Form informatischer Bildung! Zur Vereinfachung der Arbeit mit dem *computus* wurden Rechenhilfsmittel erdacht, später wurde er auch in Maschinen im Sinne von Rechenmaschinen implementiert. Die ganze Entwicklung war nicht spannungsfrei, in vielem erinnern die Wechselwirkungen des *computus* mit der mittelalterlichen Gesellschaft an die Zusammenhänge zwischen „Informatik, Mensch und Gesellschaft“ in heutiger Zeit! Und deren Erkenntnis und Bewertung wird ja ausdrücklich in den Bildungsstandards Informatik gefordert [GI08, S. 12]! Im Folgenden wird dargestellt, was die Begriffe *computus* und Computer sachlich miteinander verbindet, wie viel Mittelalter also in unserer Informationsgesellschaft enthalten ist.

2 Vom Problem zum Programm: Wann wird Ostern gefeiert?

2.1 Wurzeln der Informationstechnik

Ein Thematisieren des mittelalterlichen *computus* in der informatischen Bildung – vielleicht projektorientiert in Kooperation mit dem Geschichtsunterricht – kann auch dazu beitragen, das Bild der oft als geschichtslos dargestellten Informatik zu korrigieren! Denn informationsverarbeitende Systeme, wie wir sie heute kennen, wurden erst durch drei Entwicklungen der letzten Jahrtausende der Menschheitsgeschichte möglich [Ha96]:

- (1) *Entwicklung des Zahlensystems und des Umganges mit Zahlen (Numerik)* sowie darauf aufbauend die Entstehung und Nutzung von Mathematik und Logik.
- (2) *Aufklärung und Rationalität sowie Wissenschaft* haben die Voraussetzung dafür gebildet, dass kognitive Prozesse überhaupt verstanden und damit abbildbar gemacht werden konnten. Ohne das Prinzip der Rationalität und seiner Anwendung in einer auf Aufklärung zielenden Wissenschaft sind informationstechnische Systeme nicht denkbar. Von zentraler Bedeutung hierbei ist das explizite Herausstellen von Daten und Prozeduren, die auf diese Daten angewandt werden.
- (3) *Leistungsfähige Mechanisierung und schließlich Industrialisierung*, die dazu geführt haben, dass der Mensch bewusst eine Arbeitsteilung zwischen Mensch und Maschine entwickelt und akzeptiert hat.

Im Mittelalter unterstützten das Vorhandensein einer elementaren Numerik und die sich entwickelnde Scholastik als Vorläufer des Rationalismus das Voraus- und Zurückberechnen der Positionen der Gestirne, insbesondere von Sonne, Mond und Planeten. Das war zum einen für das Alltagsleben von hoher Bedeutung, hing doch von ihren Positionen nach damaliger Vorstellung das Schicksal der Menschen ab. Zum anderen hatte auch die Kirche Interesse an genauen Berechnungen der Positionen von Sonne und Mond, hier jedoch zur Festlegung des Kalenders und insbesondere zur Bestimmung des Osterdatums.

2.2 Numerik und die Herkunft des Wortes *computus*

Die heidnischen Römer besaßen das Tätigkeitswort *computare* für „zusammenrechnen, an den Fingern abzählen“, das daran erinnerte, dass die römischen Zahlzeichen den menschlichen Fingern nachgebildet waren. Später kam ein Wort für den Umgang mit Rechensteinen hinzu, *calcularē*, „mit Zahlzeichen rechnen“. *Calx* (Kieselstein), *calces*, *calculi* bezeichneten die Rechensteine der Römer. Ein eigenes Wort *computus* fand seit dem 4. Jahrhundert Verbreitung, eingeführt wurde es um 335 von Julius Firmicius Maternus in einem Lehrbuch der Astrologie. „Derselbe Geist, der aus dem Himmelsfeuer aufbrach und sich zur Leitung und Lenkung in das irdisch Gebrechliche einließ, überlieferte uns diese Wissenschaft, die *computi*. Er zeigte uns von Sonne, Mond und sonstigen Sternen ... den Lauf und den Rückweg, die Häuser, die Konjunktionen, die Zuwächse, Aufgänge und Untergänge“ [Bo04, S. 25]. *Computus* hieß somit nicht wie *computatio* generell Zählung oder Schätzung, sondern hatte zu tun mit der Beobachtung und Vorausberechnung von Planetenbahnen und deren astrologischer Ausdeutung.

2.3 Rationalität und die Sorgen der Skeptiker

Nach dem politischen Sieg des Christentums im Römerreich des 4. Jahrhunderts mussten verschiedene Zeitsysteme miteinander verbunden werden: der zu Neujahr beginnende Sonnenkalender der Römer (Julianischer Kalender), das jüdische Passahfest am ersten Frühlingsvollmond (also der Mondkalender) und der sonntägliche Wochenbeginn als Tag von Christi Auferstehung. Konstantin der Große verfügte 321 den Sonntag für Arbeitsruhe und Gottesdienst. Dann versuchten Kaiser und Konzil in Nicaea 325 mit der Einheit des Glaubensbekenntnisses auch das einheitliche Osterfest durchzusetzen, mit deutlicher Frontstellung gegen jüdische Bräuche. Das Konzil legte fest – zumindest nach der Darstellung späterer Computisten – dass Ostern jeweils auf den 1. Sonntag nach dem Frühlingsvollmond fallen sollte. Für die Tag-und-Nachtgleiche im Frühling, das sog. Frühlingsäquinoktium, wurde der 21. März festgelegt. Allerdings sind die erhaltenen Akten des Konzils unvollständig, im erhaltenen Rest gibt es keinerlei Beschlüsse bzgl. des Termins für das Äquinoktium oder gar einen verbindlichen Osterzyklus [Sp02, S. 36]. Es ist gut vorstellbar, dass das Konzil gar keine konkreten Beschlüsse bzgl. des Ostertermins getroffen hat, denn die frömmsten Denker der frühen Christenheit waren gegen die Verdinglichung der Zeit, somit auch gegen die kalendarische Jahresordnung und auch von den konkreten Zahlen hielten sie wenig. Langfristige Berechnungen könnten ja leicht die Versuchung fördern, sich in Gottes Pläne einzumischen und kurzfristige Messungen gaukeln dem Menschen nur die Verfügung über den Augenblick vor. Augustinus (354 bis 430), einer der bedeutendsten Kirchenlehrer an der Schwelle zwischen Antike und Mittelalter warnte: „Der gute Mensch soll sich hüten vor den Mathematikern und all jenen, die leere Vorhersagungen zu machen pflegen, schon gar dann, wenn diese Vorhersagen zutreffen. Es besteht nämlich die Gefahr, dass die Mathematiker mit dem Teufel im Bunde den Geist trüben und den Menschen in die Bande der Hölle verstricken.“ [Na01, S 23]

Und auch 1600 Jahre nach Augustinus warnen Skeptiker vor den dunklen Seiten von Wissenschaft und insbesondere Informationstechnik, etwa Joseph Weizenbaum (1923 bis 2008), der Dissident und Ketzer der Informatik (Selbstbezeichnung Weizenbaum), wenige Wochen vor seinem Tod in der Süddeutschen Zeitung: „Es ist im Prinzip unmöglich, den Menschen rein wissenschaftlich zu begreifen. Deswegen ist das Streben, Roboter in Menschengestalt herzustellen, absurd. Es kann nur aus Größenwahn oder Uterusneid entstehen. Der Glaube, dass Wissenschaft und Technologie die Erde vor den Folgen des Klimawandels bewahren wird, ist irreführend. Nichts wird unsere Kinder und Kindeskinde vor einer irdischen Hölle retten.“ [We08] All diesen Skeptikern scheint das Grundprinzip aller Informationstechnik Sorge zu bereiten, dass Maschinen (oder bei Augustinus mathematische Verfahren) aus eingegebenen Daten bearbeitete Daten erzeugen, die vom Menschen als „Information“ verstanden werden. Dieses Grundprinzip bedingt, dass informationstechnische Systeme keine Werkzeuge sind, da ja keine Materie unter Einsatz von Energie umgewandelt wird, wie das sonst Werkzeugen üblich ist. Vielmehr stellen Rechenmaschinen *Denkzeuge* dar, welche es erlauben, das Denken zu unterstützen, zu ergänzen oder auch ganz zu substituieren. Und diese Möglichkeit, das Denken Prozessen zu überlassen, die in Maschinen ablaufen, erzeugt natürlich Unsicherheit und Unbehagen in der Gesellschaft.

2.3 Der Große Osterzyklus und das Entstehen einer Wissenschaft

Dass das Mittelalter zum Zeitalter des *computus* wurde, ist dem skytischen Abt Dionysius Exiguus (der Kleinwüchsige) zu verdanken. Er entdeckte den sog. *Großen Osterzyklus* 525 (200 Jahre nach dem Konzil in Nicaea) in Rom, als er vom Papst den Auftrag erhielt, den Ostertermin des nächsten Jahres auszurechnen. Wie schon dargestellt, müssen zur Ermittlung des Osterdatums der Beginn des Naturjahres (Frühlingsanfang), das Mondjahr (Vollmonddatum) und der Wochentag (Sonntagsberechnung) in die Rechnungen einfließen. Die drei Bedingungen ergeben jährlich eine neue von Sonne und Mond abhängende Konstellation. Bereits Meton von Athen fand 433 v. Chr., dass 235 Mondumläufe fast genau in 19 Sonnenjahre passen. Daraus entstand der Alexandrinische Kalenderzyklus (19-Jahre-Zyklus), der auch heute noch die Goldene Zahl des christlichen Kalenders bestimmt. Da das julianische Jahr alle vier Jahre einen Schalttag aufweist, ergibt sich für die Wochentage ein Sonnenzyklus von 7×4 julianischen Jahren, nach dem sich die Zuordnung der Wochentage zu den 365 Tagen des Jahres exakt wiederholt. Mit dem Mondzyklus zusammen entsteht ein großer Zyklus, der sog. Osterzyklus, von $28 \times 19 = 532$ Jahren, nach dessen Ablauf die Ostertermine in gleicher Ordnung wiederkehren. Dionysius stellte eine Tafel auf (Abbildung 1), die 532 Jahre umfasst. Dafür musste natürlich ein Bezugspunkt gesetzt werden. Dionysius bezog seine Ostertafel auf das Anfangsdatum der *incarnatione domini nostri Jesu Christi*, der Fleischwerdung von Jesus Christus. Wenn Christus Herr über die Zeit war, durfte man sich sein Erdendasein auch in die Zeitspirale holen! Seit dieser Zeit gibt es also die heute gebräuchlichen Jahreszahlen, die allerdings erst sehr viel später auch wirklich verwendet wurden.



Abbildung 1: Ein frühes Rechenhilfsmittel, die Ostertafel von Ravenna, mit dem 532jährigen Osterzyklus.

Cassiodor, ein Zeitgenosse von Dionysius, preist um 550 die Arithmetik als eine Grundwissenschaft: „Es ist uns auch gegeben, größtenteils unter Anleitung dieser Disziplin zu leben. Wenn wir durch sie die Stunden lernen, wenn wir die Monatsläufe ausrechnen, wenn wir den Zeitraum des wiederkehrenden Jahres berechnen, werden wir durch die Zahl belehrt und vor Verwirrung bewahrt. Nimm der Welt den *compotus*, und alles verfällt blinder Unwissenheit“ [Bo04, S. 32]. Aus dem Kreis um Cassiodor kam 562 die erste Schrift mit dem programmatischen Titel „*Computus paschalis*“. Ostertag und Zeitrechnung waren darin miteinander verzahnt, *computus* hieß fortan Osterberechnung. Isidor von Sevilla sammelte um 630 die kalendarischen Formeln und stellte sie zur freien Verfügung der Christenheit. Obgleich auch *computare* bei Isidor einfach addieren oder multiplizieren heißen konnte, erhoben sich diejenigen, welche universaler Zeitrechnung nachgingen, weit über den Kleinkram eines *calculator*, der einzelne Zahlen zusammenklaubte. Der *computus* war damit eine Sammlung von nicht trivialen Rechenvorschriften – heute würden wir Algorithmen dazu sagen – die es ermöglichten, zu einem bestimmten Jahr das Osterdatum, die darauf bezogenen kirchlichen Feiertage und eigentlich den gesamten Jahreskalender anzugeben. Im Unterschied zur algebraischen Handhabung (zum Beispiel bei der Bestimmung des größten gemeinsamen Teilers von Zahlen nach Euklid) handelte es sich beim *computus* um einen *Prozess*, der Algorithmus schließt also die Beschreibung des zeitlichen Ablaufs der einzelnen Rechenschritte mit ein! Damit ist der *computus* vergleichbar mit der genau determinierten Webstuhlsteuerung mittels Lochkarten, der Steuerung eines Musikautomaten durch die Stiftwalze oder den Verfahren zur Ver- und Entschlüsselung von Daten! Der *computus* ermöglichte damit schon, wofür unsere Computer heute auch oft genutzt werden: Simulationen. Die theoretische Grundlage der Planetenbewegung, das Gravitationsgesetz, war noch nicht entdeckt. Mit dem *computus* stand ein Modell zur Verfügung, mit dem sich Prozesse durchspielen ließen, hier eben die Bewegung von Sonne und Mond.

3 Der Computus im Lauf der Jahrhunderte

3.1 Komputistik: Eine Wissenschaft entwickelt sich

Die Ostertafel des Dionysios aus dem Jahr 525 musste für die Jahre ab 722 neu berechnet werden, was für eine Vereinheitlichung hätte sorgen können. Allerdings vertrauten Zeitrechner wie Isidor von Sevilla und dessen Schüler nicht dem Griechen Dionysios sondern eher dem Gallier Victorius von Aquitanien, der schon im Jahr 457 für den römischen Papst eine „lateinische“ Ostertafel mit einem 84jährigem Zyklus errechnet hatte, die auch noch im 8. Jahrhundert ohne Änderung zu benutzen war. Und so liefen für ein und dasselbe Verfahren der Zeitbestimmung die verschiedensten Formeln, Begriffe und Erklärungen um. All das musste gesammelt und verglichen; darüber hinaus Experten ausgebildet werden, die die verschiedenen Meinungen kompetent vertreten konnten. Diese Experten erhielten in einem merowingischen Lehrbuch von 727 erstmals die passende Berufsbezeichnung, *computistae*. Diese benahmten sich nun so, wie es sich für Experten gehört: Kaum waren sie unter sich, begannen sie, sich zu zanken. Streiten sich heute die Informatiker um Betriebssysteme oder Programmiersprachen, so stritten sich deren Vorläufer, die Komputisten, um die richtigen Algorithmen zur Berechnung des Osterdatums. So kam es zwar zu einer raschen Verschriftlichung der Komputistik und zur Verbreitung ihrer Ideen, von einer Vereinheitlichung konnte aber keine Rede sein. Nun musste aber das Osterfest zum überall gleichen Zeitpunkt begangen werden, nicht in Irland so und bei den Römern anders! Diese Forderung bewegte einen Meister der Komputistik, den angelsächsischen Mönch Beda Venerabilis. Sein generelles Lehrbuch von 725 hieß „De temporum ratione“, Rechenschaft von den Zeiten, das erste Kapitel darin: *De computo vel loquela digitorum*. Beda entschied sich in seinem Lehrbuch gegen die Ostertafel des „Lateiners“ Victorius (84jähriger Zyklus) und für die des Griechen Dionysius Exiguus (532jähriger Zyklus). Und Beda führte 731 in seiner „Kirchengeschichte des englischen Volkes“ die Jahresdatierung nach der Fleischwerdung Christi ein. Dieses Buch wurde vorbildlich für die mittelalterliche Geschichtsschreibung, daher reden wir heute nicht vom 2761ten Jahr nach Gründung der Stadt Rom sondern vom Jahr 2009 nach Christi Geburt. Und auch der Jahreskalender, zunächst verfasst nach dem caesarischen System, profitierte von der Komputistik. Die von den Komputisten ermittelten Sternbewegungen und Rechenhilfen wurden in den Jahreskalender übertragen, damit wurde er zum Zeitweiser nicht nur für geistliche Zwecke sondern gab praktische Hilfe für das Leben auf Erden. Und auch das ist typisch für die Entwicklung informationsverarbeitender Systeme: Es entstehen bei der Arbeit an informatischen Problemen „Abfallprodukte“, die gern von Nicht-Informatikern aufgegriffen und genutzt werden. Textverarbeitungs- und Tabellenkalkulationssysteme und natürlich auch die Internetdienste seien als Beispiele der heutigen Zeit angeführt.

3.2 Komputistik und Politik

Die dargestellten Zusammenhänge wurden in der Karolingerzeit noch vertieft. So befahl Karl der Große 789, dass alle Geistlichen neben Psalmen, Noten, Grammatik mit dem *computus* vertraut und mit Lehrbüchern darüber versehen sein müssten. Der Kaiser ging mit gutem Beispiel voran, erlernte die *ars computandi* und betrieb aktiv Sternenkunde. Das Interesse Karls am *computus* hatte handfeste Gründe: Er begann nach 780 sein Reich über einen Großteil des lateinischen Europa auszudehnen. Um seine Untertanen zu beherrschen und zu vereinen, drängte Karl auf eine Vereinheitlichung der kalendarischen Praxis. Diese karolingische Kalenderreform setzte bei den theoretischen Grundlagen der Zeitrechnung an, also bei der Komputistik. Karl rief dazu Fachleute aus aller Herren Länder an seinen Hof, dass sie sich über Deutung, Bestimmung und Nutzung der Zeit einigen mögen. Damit löste er seit 790 eine neue Serie komputistischer Streitschriften aus. Die meisten Experten waren inzwischen bereit, Bedas Konzept der dionysianischen Osterrechnung (532jähriger Zyklus) zu übernehmen; die neuartige Berechnung der Weltjahre seit Christi Geburt fand indes weniger Anklang. Aber auch hier schlug Karl mit der Faust auf den Tisch; seine Kaiserkrönung Weihnachten 800 sollte mit der richtigen Zeitangabe in die Geschichte eingehen. Und Kaiser Karl ließ in einer siebenbändigen Enzyklopädie zusammentragen, was Fachleute über Komputistik wissen sollten. Eine Sammlung komputistischer Regeln lehrte die enthaltenen Tabellen und den Jahreskalender zu benutzen. Den lokalen Fürsten wird die zentrale kaiserliche Regelung des *computus* nicht recht gewesen sein, bedeutete der einheitliche Reichskalender doch einen Machtverlust zugunsten der Zentralgewalt. Vielleicht kommt aus dieser Zeit auch die Urangst heutiger föderaler Bildungspolitik, die Unterweisungen in den „modernen *computus*“, sprich den Informatikunterricht, in den Lehrplänen verbindlich zu verankern.

3.3 Die Verdrängung des *computus* durch Astrolab und Abacus

Schon hundert Jahre nach Karl machte sich eine Ermüdung in Komputistik und Chronistik breit; den meisten Geistlichen wurde nicht mehr das Studium von Bedas Grundgleichungen zugemutet, nur noch die Kenntnis des *computus minor*, der Faustregeln für das laufende Jahr waren nötig. Trotzdem forderten einzelne Bischöfe, vornehmlich im Westfrankenreich, von ihren Priestern die Kenntnis des *computus necessarius*. Sie sollten anhand von Buchstaben für Sonntage und Mondstände, von Zahlen für Epakten, Konkurrenten und Regulare die Wochentage, Monatsanfänge, Fastenzeiten, Ostertermine und Hauptfeste des Jahres bestimmen können. Im zehnten Jahrhundert jedoch konnte das Kirchenrecht nicht mehr verlangen, dass jeder Priester den *computus* beherrschen müsse. Auch das scheint ein Erbe des *computus* zu sein, nutzen doch auch heute viele Anwender den Computer lediglich nach „Faustregeln“, wie beispielsweise die Inhalte des „Europäischen Computerführerscheins (www.ecdl.de) deutlich machen. Die Kenntnis der Grundlagen von informationsverarbeitenden Systemen scheint immer weniger gefragt zu sein. Immer häufiger sind informationsverarbeitende Verfahren in Gegenstände des täglichen Gebrauchs eingebettet – wir reden auch von *embedded systems* – und nicht mehr als solche zu erkennen.

Auch diese Entwicklung gab es bereits im Mittelalter! Dieselben Araber, die Nordspanien in jener Zeit verheerten, brachten auch Kenntnisse und Geräte für Zeitmessung mit, deren Genauigkeit das christliche Europa verblüffte. So ein Gerät der Araber war das Astrolab, mit dem die Bewegung der Sonne und der Fixsterne genau gemessen werden konnte und das auch eine Umrechnung der Temporalstunden des Sonntages in die wahre Himmelszeit der Fixsternsphäre mit 24 Äquinoktialstunden erlaubte. Für diese Umrechnungen setzten sich die Begriffe *calcolare* und *numerare* durch. War ein Astrolab richtig gebaut und eingestellt, nahm seine analoge Anzeige dem Menschen Gedächtnisleistung und Rechenarbeit ab. Das Astrolab wurde damit zum frühesten Analogrechner der europäischen Wissenschaft und zum modernen Konkurrenten des *computus*!

Und auch ein Rechengerät, das „digital“ arbeitete, nämlich die Rechenergebnisse durch voneinander getrennte Zahlzeichen angab, wurde intensiv genutzt, der Abacus. Gerbert von Aurillac (um 950 bis 1003), der spätere Papst Silvester II, bezeichnete beim Multiplizieren mit dem Abacus die Zahlen des Produkts *digiti*, Fingerzahlen, soweit sie die Zahl von 1 bis 9 angaben, und *articiuli*, Gelenkzahlen, wenn sie darüber hinaus gingen. Gerbert nannte die Benutzer des Rechenbrettes *abacista*; nicht etwa *computista*. Komputistische Termini und Methoden begannen sich aufzulösen und mit allgemein mathematischen zu vermengen. Mit Astrolab und Abacus wurde die Zeitrechnung genauer, da sie sich auf Beobachtungen stützte. Der *Computus* wurde der neuen Arithmetik zugeschlagen; die Zahl löste sich von der Zeit. War der *computus* also im Begriff, von der theoretischen Klärung der Zeitbegriffe abzulassen und in Rechenpraxis aufzugehen?

3.4 Algorismus und *computus ecclesiasticus*

Im späten 12. Jahrhundert erlebte die Zeitrechnung einen unerwarteten Aufschwung. Die neue Rechenpraxis des Dezimalsystems, der nach al-Chwarizmi's Rechenbuch benannte *Algorismus* drang in die Komputistik ein, früher als in die königliche Finanzpolitik oder bürgerliche Geldwirtschaft. In einem „Salzburger *Computus*“ stellte 1143 ein Anonymus das Verfahren der Zeitrechnung auf indische Ziffern um, während sich Geschäftsleute noch zweihundert Jahre mit den sperrigen römischen Zahlen abmühten [Bo04, S. 84]. Mit indischen Ziffern ermittelten die Komputisten die entscheidenden Bruchteile bei der Berechnung des Mondmonats. In den 1250er Jahren wurde der erreichte Fortschritt im Umgang mit Zahlen gewürdigt und gefordert, Zeitrechnung mit indischen Ziffern zu betreiben. Einer der zentralen Begriffe des Informatikunterrichtes, der „Algorithmus“, ist also schon seit frühester Zeit eng mit dem *computus* verbunden! Der größte Enzyklopädist des Hochmittelalters, Vinzenz von Beauvais, besprach beide Konzepte zusammen „De computo et algorismo“ [Bo04, S. 88]. Es lag nahe, die neuen Methoden nun auch zur Verbesserung der christlichen Zeitrechnung heranzuziehen. Aber nichts dergleichen geschah. Die Scholastiker Albertus Magnus und Thomas von Aquin wussten, dass sich Zeit mit modernen Verfahren genau errechnen und beobachten ließ. Das herkömmliche und nicht genaue Verfahren nannten sie *computus ecclesiasticus*, ein Begriff, der bis in unsere Zeit überdauert hat, wie später gezeigt wird.

3.5 Vom computus zum Konto

Die Engländer formten den *computus* noch entschiedener um. König Heinrich I. überließ um 1130 den Londoner Bürgern einträgliche Verwaltungsämter gegen Entrichtung von 300 Pfund *ad computum*, also gegen rechnerischen Nachweis, vor der königlichen Finanzbehörde. Schatzmeister Richard von Ely beschrieb um 1178 die Arbeit dieser Behörde und erwähnt nicht nur den *calculator*, der die Summen am Abacus verrechnet, sondern auch *computatores*, die das eingehende Bargeld zählen. Die jährlichen Schlussabrechnungen heißen bei Richard *magni annalis compotorum rotuli*. Das Geldwesen verlagerte den Zusammenhang zwischen Zeit und Zahl ganz ins Stoffliche. In Italien hieß *conto* noch um 1250 astronomische Zeitrechnung, wie lateinisch *computus*, aus dem es abgeleitet war. Zusammen mit der Buchführung wandert das Wort *conto* ins Französische als *compte*, ins spanische als *Cuento* und ins Deutsche als *Konto*. Die Finanzwirtschaft drang wie bei den Engländern in die öffentliche Verwaltung ein. In Paris hieß die Rechnungskammer in der Mitte des 13. Jahrhunderts *camera compotorum*, die Deutschen entlehnten im 15. Jahrhundert *comptoir* für Zahlstisch, später Handelsniederlassung; im Kontor lebt es heute noch fort. Die Wanderung des Begriffs *computus* in das Finanzwesen hatte gute Gründe: Kalendertage müssen hier klar definiert und addiert werden können. Die Praxis der Benennung von Tagen nach christlichen Heiligen, wie wir sie auch heute noch in den Namenstagen finden, hätte zu Unklarheiten geführt. In den verschiedenen Landstrichen wurden die Tage eines Jahres durchaus jeweils anderen Heiligen geweiht.

4 Der Computus implementiert

4.1 Mechanisierung: Uhren und Rechenmaschinen

Haben sich die Verfahren zur Berechnung des Osterdatums auch in Geräten niedergeschlagen, so dass man von Vorläufern des *computus* im Sinn von Rechenmaschinen reden kann? Diese Frage ist für die Informatik von grundlegender Bedeutung, definiert sie sich doch als Wissenschaft von der systematischen Verarbeitung von Informationen, insbesondere der automatischen Verarbeitung mithilfe von Rechanlagen [CS88, S. 269]. Die angeführten astronomischen Phänomene von wiederkehrenden Ereignissen, die Umläufe von Erde, Mond, Sonne und Sternen, lassen sich besonders gut in mechanisierten, zyklischen Abläufen darstellen. Mechanische Geräte wie Uhren erlauben eine solche Darstellung aus ihrer Natur heraus. Die analogen Uhren mit Ziffernblatt überstreichen in 24 Stunden zweimal die 360 des Kreises mit dem Stundenzeiger. Bereits seit dem 14. Jahrhundert wurden planetarische Uhren entwickelt, die Abläufe darstellen, unter anderem die Mechanismen des *computus*. Eindrucksvolle Beispiele dafür sind in Mitteleuropa die astronomischen Uhren des Straßburger Münsters, der Rostocker Marienkirche oder des Altstädter Rathauses in Prag. Das 17. Jahrhundert baute Rechenschieber und Rechenmaschinen, mit denen auch die Zeit berechnet werden konnte.

Die erste moderne Rechenmaschine, entworfen 1623-24 von dem Tübinger Mathematiker Wilhelm Schickard, sollte die chronologischen und astronomischen Arbeiten von Johannes Kepler unterstützen. Und auch Schickard benutzt das Wortfeld des *computus*, wenn er über seine Maschine schreibt, sie könne sechsstellige Zahlen automatisch zusammenrechnen, *datos numeros statim automatos computet* [Bo04, S. 118]. Ganz anders der junge Blaise Pascal, der für seinen Vater, einem Steuerpächter, ebenfalls eine Rechenmaschine erfand. Obwohl sie ihm ein Uhrmacher nach seinen Plänen baute, empfahl er ihre Benutzung nicht den Zeitrechnern. Pascal wandte sich gegen die rationalistische Mode seines Jahrhunderts, die unter dem Einfluss von Descartes den Menschen als Mechanismus aus Leib und Seele verstand und darum für berechenbar hielt. Für Pascal waren Weltschöpfung und Heilsgeschichte keine Rechenaufgaben sondern Glaubenssache. Mit dieser Einstellung wollten viele Europäer, die im 17. und 18. Jahrhundert Rechenmaschinen erfanden, lediglich den Geist hervorragender Denker von mechanischer Routinearbeit befreien. Die Geräte wurden daher lateinisch als *machina arithmetica*, englisch als *Calculating Machine* und deutsch als *Rechnungs-Maschine*, später *Rechenmaschine* bezeichnet. Für komputistische Zwecke wurde keine dieser Maschinen gebraucht.

4.2 Der Kirchenkomput im Straßburger Münster

Trotzdem gab es erfolgreiche Versuche, die mittelalterliche Komputistik in mechanischen Geräten fortleben zu lassen. Die astronomische Uhr des Straßburger Münsters ist wohl eines der eindrucksvollsten Beispiele für die Implementierung des *computus* weltweit. Die heutige astronomische Uhr im Münster, die bisher dritte an dieser Stelle, ist von 1838 bis 1848 durch Jean Baptiste Schwilgué (1776 bis 1856) erbaut worden. Er hatte die zweite Uhr als Kind kennengelernt und erlebte 1788, dass sie nicht mehr in Bewegung zu setzen war. Schwilgué eignete sich im Eigenstudium die notwendigen Kenntnisse an, um die Uhr zu modernisieren und in der heute noch zu besichtigenden Form zu realisieren. Als schwierigstes Problem erwies sich der sog. Kirchenkomput. Er vereinigt alle zyklischen Daten wie Jahr, Sonnenszirkel, Goldene Zahl, Indiktion, Sonntagsbuchstabe und Epakten (Abbildung 2). Das Gerät ist in der Lage, die Zyklen, Epakten, Sonntagsbuchstaben und Ostersonntage für alle Jahre seit der Gregorianischen Reform 1582 anzugeben. Sechs symmetrische Zifferblätter stellen die Daten des *Computus* dar. Die Jahreszahl steht oben, in der Mitte wird die *Indiktion* (Römische Zinszahl) angezeigt, die noch heute für die Datierung päpstlicher Bullen Verwendung findet. Sie verfolgt einen Zyklus von 15 Jahren. Der 28jährige Sonnenzyklus wird oben links angezeigt. Unten links findet sich der Sonntagsbuchstabe. Er weist jedem Tag des Jahres einen Buchstaben zu, der sich zyklisch verschiebt. Die Goldene Zahl wird oben rechts berechnet. Sie durchläuft die Werte von 1 bis 19. Unten rechts stehen schließlich die Epakten, die für den Gregorianischen Kalender erfunden werden mussten.



Abbildung 2: Der Kirchenkomput im Straßburger Münster.

Die Regeln des Gregorianischen Kalenders sowie die inkommensurablen Umlaufzeiten für Sonne und Mond machten eine mechanische Realisierung des *computus* fast unlösbar. Trotzdem gelang dem Mechaniker Schwilgué 1815 die Lösung! Die Berechnung der Zahnräder ermöglicht eine Genauigkeit für ein mittleres Erdjahr von zwei Millisekunden! Alle Anzeigen werden auch in 1000 Jahren noch die genauen astronomischen Ereignisse melden, wenn es gelingt, die Mechanik so lange in Funktion zu halten! Dieser frühe Computer stellt mit seiner Korrektheit des Programms sowie seiner Zuverlässigkeit wohl alle Rechner heute weit in den Schatten! Natürlich ist das alles mechanisch fest programmiert. Aber auch gerade heute arbeitet die Mehrzahl der Computer in Handys, Autos, Waschmaschinen usw. nicht nach der frei programmierbaren Weise der digitalen Universalrechner, sondern mit ebenfalls festem Programm. Für sie alle ist der *computus* des Mittelalters der Namensgeber und die astronomische Uhr in Straßburg die noch immer funktionsfähige Implementation.

4.3 Industrialisierung und die Wiedergeburt des *computus*

Das 19. Jahrhundert stellte Rechenmaschinen als Massenware her und nannte sie meistens *Calculating Machines*. Sie rationalisierten jene Rechenarbeiten in Wirtschaft und Verwaltung, die wenig Intelligenz aber viel Zeit und damit Lohn kosteten. Im Jahr 1878 wurde jedoch eine mechanische Rechenmaschine des Amerikaners William Hart unter dem Namen *Hart's Mercantile Computing Machine* patentiert (siehe Abbildung 3), deren runde Bauform wieder an die zyklischen Berechnungen des *computus* erinnert.



Abbildung 3: Hart's Mercantile Computing Machine, 1878.

Der amerikanische Ingenieur und Statistiker Herman Hollerith baute seit 1884 ein *Electric Tabulating System* mit uhrenförmigen Zählwerken, *Counters*. Diese Maschine vereinfachte 1890 die Volkszählung in den USA auf der Basis von Lochkarten. Die erste wirklich brauchbare und wirtschaftlich erfolgreiche Addiermaschine war das *Comptometer*, entwickelt vom Amerikaner Dorr Eugene Felt und patentiert im Jahr 1887. Mit Holleriths *Counters*, der *Computing Machine* von Hart oder dem *Comptometer* gelangte also das Wortfeld des *computus* zu neuen Ehren. Wendungen wie Computer oder to compute waren allerdings für Menschen und deren höhere Mathematik reserviert. Diese Grenze wurde kurz vor der Jahrhundertwende überschritten. In einer englischen Ingenieurzeitschrift wurde 1897 erstmals der Name Computer für Rechenmaschinen benutzt. Zufällig wurde im selben Jahr Papst Leo XIII. von Astronomen vorgeschlagen, die Komputistik durch Einführung eines immerwährenden Kalenders mit festem Osterdatum ganz abzuschaffen. Holleriths Lochkartenmaschine hieß wenige Jahre später *Statistical Computer*; seine Firma, die jetzige IBM, seit 1911 *Computing Tabulating Recording Company*. Als in den dreißiger und vierziger Jahren Maschinen zur elektronischen Datenverarbeitung entwickelt wurden, verkürzte man das übliche *Calculation Machine* zu *Calculator*; analog wurde seit 1940 die *Computing Machine* zum Computer. Auf die Unterscheidung zwischen dem Menschen und seinem Denk-Zeug wird vollends verzichtet.

An der Genese des Begriffs *computus* lassen sich in der Schule wichtige Etappen der Entwicklung der Informatik deutlich machen: Die Verarbeitung von Informationen begann mit der Entwicklung mathematischen Denkens (zyklische Bewegungen der Himmelskörper, 532jähriger Osterzyklus). Unter Einbeziehung von gegenständlichen Hilfsmitteln (Ostertafel) sowie von mechanischen, elektrischen oder elektronischen Prinzipien lassen sich die jeweiligen Operationen erleichtern (Astrolab, Abacus), beschleunigen (Schickardsche Rechenmaschine) und weitestgehend fehlerfrei realisieren (Kirchenkomput). Weiterhin konnte der Grad der Verknüpfung von Operanden erheblich

erweitert werden (Kirchenkomput). Im *computus* zeigt sich bereits das funktionale Schema der programmgesteuerten Rechenautomaten, die Denkprozesse des Menschen weitestgehend nachzubilden. Zu Recht tragen daher unsere heutigen Rechenmaschinen seinen Namen.

Literaturverzeichnis

- [Bo04] Borst, Arno: *Computus - Zeit und Zahl*. Wagenbach, Berlin, 2004.
- [CS88] Claus, V.; Schwill, A: *Duden Informatik*. Dudenverlag, Mannheim, 1988.
- [GI08] Gesellschaft für Informatik e. V.: *Grundsätze und Standards für die Informatik in der Schule*, Berlin, 2008.
- [Ha96] Haefner, Klaus: *Informationstechnik und ihre Nutzung in der Gesellschaft*. PDF-Dokument, 1996, <http://www2.tu-berlin.de/~dibpaed/LehrveranstaltunglHomepage-Stiehl-Stand021107/ITG-M.htm>.
- [Na01] Naumann, Friedrich: *Vom Abakus zum Internet – Die Geschichte der Informatik*. Primus-Verlag, Darmstadt, 2001.
- [Sp02] Springfeld, Kerstin: *Alkuins Einfluss auf die Komputistik zur Zeit Karls des Großen*. Stuttgart, 2002.
- [We08] Weizenbaum, Joseph: *Grenzen des Wissens – Wir gegen die Gier*. In: *Süddeutsche Zeitung*, 8.1.2008, <http://www.sueddeutsche.de/wissen/529/429282/text/>.

Informatikdidaktik – Einschätzung der Landschaft

Ludger Humbert

Abstract: Im Mai 2009 sind in der Bundesrepublik Deutschland sechzehn Fachdidaktikprofessuren Informatik besetzt. An mindestens 35 Hochschulen in der Bundesrepublik kann mit dem Studienziel »Lehramt Informatik« studiert werden. Das Fachgebiet »Didaktik der Informatik« beginnt mit seiner stärkeren Ausdifferenzierung – es kann von ersten Fachdidaktikschulen gesprochen werden. In diesem Beitrag wird der Versuch unternommen, ein wenig Ordnung in die Informatikfachdidaktiklandschaft zu bringen

Kriterien des Ratings werden darauf hin untersucht und beurteilt, ob sie es erlauben, die Landschaft der Fachdidaktik Informatik einer strukturierten Analyse zugänglich zu machen. Der Beitrag soll dazu anregen, sich der Aufgabe der Evaluation stärker zu stellen – die Fachdidaktik Informatik sollte es als originäre Aufgabe ansehen, ihre Ergebnisse intern zu evaluieren und einer externen Evaluation zugänglich zu machen.

Der nachweisbaren Dissemination von Forschungsergebnissen in den konkreten Informatikunterricht – und damit bis zur konkreten Schülerin – kommt in den Fachdidaktiken eine Schlüsselrolle zu. Diese Dimension findet bei den gängigen Formen des [Forschungs-]Ratings keine Beachtung. Daher müssen für die Fachdidaktik Informatik andere Formen des Ratings entwickelt und erprobt werden.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21. - 24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Der Bologna-Prozess in der Lehrerausbildung aus dem Blickwinkel der Informatik – Vorschläge zur Konsolidierung einer laufenden Reform

Hanno Schauer

Lehrstuhl für Wirtschaftsinformatik und Unternehmensmodellierung
Institut für Informatik und Wirtschaftsinformatik (ICB)

Universität Duisburg-Essen

Universitätsstr. 9

D-45141 Essen

hanno.schauer@uni-duisburg-essen.de

Abstract: Die Mehrzahl der deutschen Bundesländer führt die im Bologna-Prozesses verabredeten Strukturen der Hochschulausbildung auch für das Lehramtsstudium ein. Die Neuorganisation der Studiengänge erfolgt hierbei weitgehend länderspezifisch. Im Ergebnis wird die Landschaft der Lehrerausbildung in Deutschland – im Kontrast zum Geist von Bologna – nicht einheitlicher, sondern heterogener. Das vorliegende Diskussionspapier erörtert mit Blick auf die Bologna-Ziele die eingeschlagenen Reformen für das Lehramt allgemein und speziell für die Ausbildung von Informatik-Lehrern. Auf dieser Grundlage werden Vorschläge zur Konsolidierung der begonnen Reformen entwickelt.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21. - 24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Dualitätsrekonstruktion als Hilfsmittel zur Entwicklung und Planung von Informatikunterricht

Carsten Schulte
Institut für Informatik
Freie Universität Berlin
Königin-Luise-Str. 24-26
10495 Berlin
schulte@inf.fu-berlin.de

Abstract: Bislang gibt es kein informatikdidaktisches Verfahren für die didaktische Rekonstruktion von digitalen Artefakten wie z.B. Standardsoftware, um sie für die Unterrichtsplanung aufzubereiten. Im Artikel wird in Form der Dualitätsrekonstruktion ein solches (einfach zu handhabendes) Verfahren vorgeschlagen, das insbesondere auf drei Aspekte zielt: Aktuelle Gegenstände und Beispiele für den IU zugänglich machen; schülerorientierte Unterrichtsplanung und -gestaltung, Verdeutlichung des Alltagsbezugs informatischer Bildung.

1. Überblick

Im Artikel wird ein einfaches Verfahren vorgestellt, mit dessen Hilfe digitale Artefakte [Sc08] wie beispielsweise Handys, MP3-Player, Tabellenkalkulation, digitales Video o. Ä. für den Informatikunterricht didaktisch aufbereitet (rekonstruiert) werden können. Damit können Alltagserfahrungen aufgegriffen und der Alltagsbezug informatischer Bildung verdeutlicht werden. Insbesondere sollen die informatischen Hintergründe so aufgezeigt werden, dass sie als sinnvoll erlebt werden können, indem sie die Funktionsweise digitaler Artefakte erklären, zur Bewältigung komplexer Nutzungsszenarien beitragen und die Lücke zwischen Alltagserfahrungen mit digitalen Artefakten und der Welt der Informatik überbrücken helfen.

Den zentralen Ausgangspunkt der didaktischen Rekonstruktion bildet die Betrachtung der dualen Natur digitaler Artefakte. Die beiden Seiten der Dualität sind Struktur und Funktion. Funktion bezieht sich auf den Einsatzzweck und die Nutzung des Artefakts, die Struktur auf dessen inneren Aufbau. Die entsprechende Analyse wird angereichert durch die Rekonstruktion des (geschichtlichen) Entwicklungspfads, der wesentliche konzeptuelle Ideen und die dynamische Veränderbarkeit aufzeigen hilft. Diese Schritte werden ergänzt durch die Analyse bzw. Rekonstruktion von Lernvoraussetzungen wie Präkonzepten und Vorerfahrungen der Lernenden sowie die Definition der beabsichtigten Lernziele. Lernvoraussetzungen und Lernziele werden jeweils auf den drei Dimensionen Weltbild, Selbstbild und Verhaltensweisen beschrieben.

Bevor der theoretische Hintergrund dieser Konzeption ausgeführt wird, zeigt die folgende Abbildung zunächst ein einführendes Beispiel einer entsprechenden didaktischen Rekonstruktion. Das Verfahren soll auf einfache Weise (daher die Darstellung als Mindmap) die Ideen- und Themenfindung und das Herausarbeiten der inneren Verknüpfung verschiedener wesentlicher didaktischer Dimensionen unterstützen: Fachlicher Gehalt, Vorwissen und Interessen der Lernenden sowie das Bestimmen der Lernziele. Die Abbildung zeigt den grundsätzlichen Aufbau der didaktischen Rekonstruktion, stellt jedoch eine naive Variante dar.

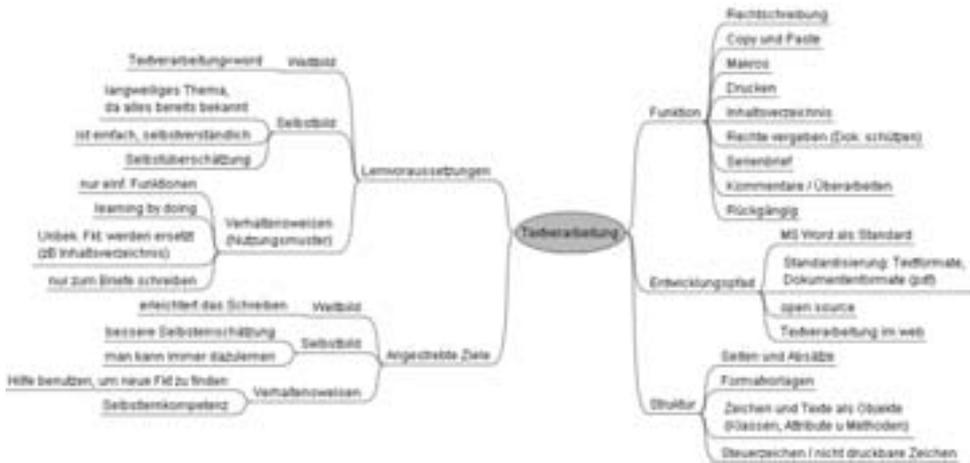


Abbildung 1: Naive didaktische Rekonstruktion der Textverarbeitung

Dieses naive Herangehen birgt zwei Probleme: Erstens wird die Rekonstruktion von Struktur und Funktion schwierig sein und daher wie oben recht oberflächlich bleiben. Das gilt, zweitens, analog für die Bestimmung von Lernzielen: Was sollen denn die Schülerinnen und Schüler überhaupt noch lernen, wenn sie sich vermutlich sogar besser mit der Nutzung auskennen als der Lehrer?

Die folgenden Kapitel ergänzen notwendige theoretische Hintergründe zu Lernvoraussetzungen und zur Dualität digitaler Artefakte, mit denen die Rekonstruktion schrittweise geschärft und ergänzt wird. Dabei wird jeweils auch das einführende Beispiel ergänzt werden.

2. Die duale Natur digitaler Artefakte

Der Ansatz stützt sich auf den technikphilosophischen Ansatz der dualen Natur technischer Gegenstände, demzufolge diese nur dann vollständig erfasst und erklärt werden können, wenn sowohl deren Struktur und Funktion als auch der Zusammenhang dieser beiden Dimensionen berücksichtigt wird. Strukturelemente beziehen sich auf den Aufbau des Artefakts, beispielsweise verwendete informatische Konzepte, Algorithmen und Datenstrukturen. Diese Dimension kann objektiv gemessen werden. Die andere Seite der Dualität, die Funktion, bezieht sich auf den Einsatzzweck des Artefakts und beschreibt

dessen potenziellen Nutzen. Dieser kann jedoch nicht objektiv gemessen werden. Vielmehr handelt es sich um eine sozial verankerte Zuschreibung. Die Dualität von Struktur und Funktion bewirkt, dass zumeist entweder nur die Funktion aus Benutzersicht bzw. von außen, oder die technische Struktur aus Entwicklersicht bzw. von innen, betrachtet wird. Die Funktion betrifft den Nutzen, den das Artefakt bietet, dessen Funktionalität und Eignung, damit die spezifischen individuellen Aufgaben zu erledigen bzw. Einsatzzwecke zu erreichen. Sie antwortet auf die Frage: Was kann ich damit machen? Die Struktur betrifft die physikalische, technische, informatische Seite: Wie ist das Artefakt aufgebaut, zusammengesetzt und wie funktioniert es, welche Prozesse laufen im Inneren ab? Die fachdidaktisch relevante Schlussfolgerung dieser Betrachtung ist, dass digitale Artefakte erstens sowohl unter der Frage des ‚Wie‘ (Struktur) als auch unter der Frage des ‚Warum‘ und ‚Wozu‘ (Funktion) zu betrachten sind; denn Struktur wird ohne Funktion buchstäblich sinnlos. Zudem kann ist Funktion auf entsprechende Struktur angewiesen, ansonsten kann sie nicht verwirklicht werden.

Die folgende Abbildung zeigt die Rekonstruktion der Dimensionen Struktur und Funktion sowie des Entwicklungspfad für die Textverarbeitung. Die Abbildung zeigt verschiedene Merkmale von Struktur, Funktion und Entwicklungspfad auf. Diese Merkmale erheben keinen Anspruch auf Vollständigkeit oder objektive Gewichtung, sondern stellen vielmehr eine an didaktischen Kriterien zu messende Gewichtung wesentliche Lerninhalte und -ziele dar.

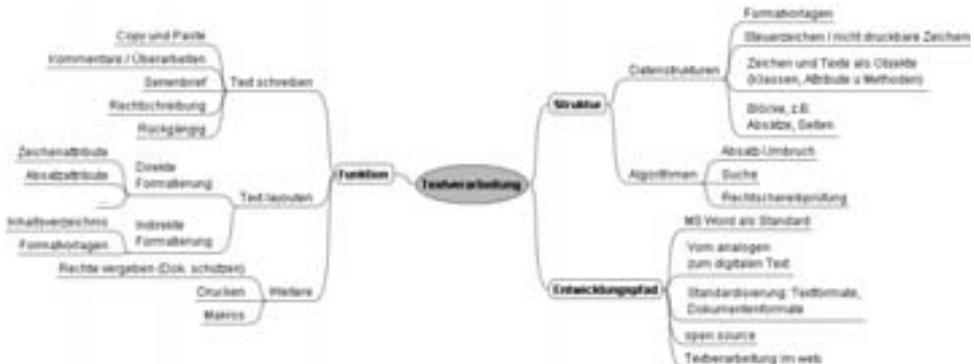


Abbildung 2: Struktur und Funktion des digitalen Artefakts Textverarbeitung

Die Beschreibung der Funktion wird durch den implizit oder explizit berücksichtigten Kontext beeinflusst. In verschiedenen Kontexten sind unterschiedliche Funktionen wichtiger, man denke etwa an Textverarbeitung im Kontext von Seminararbeiten oder in einer kleinen Firma, die Rechnungen ausstellt und Kundenkontakte pflegt (zudem handelt es sich bei der Funktion ja um eine Zuschreibung der Intention). Die in der Abbildung genannten Aspekte sind daher zu einem gewissen Grad subjektiv gefärbt und stellen einen Ausschnitt dar. Ähnliches gilt für die Beschreibung der Struktur – hier können und sollen nur die wesentlichen informatischen Ideen genannt werden. Das birgt ein wenig die Gefahr, Struktur und Funktion als die beiden unverbundenen Pole der Dualität zu fassen. Um dieser Gefahr zu begegnen und die Verzahnung von Funktion und Struktur aufzuzei-

gen, wird die Rekonstruktion von Struktur und Funktion um den Entwicklungspfad ergänzt. Dieser beschreibt die Ideengeschichte des digitalen Artefakts. Dazu werden wesentliche Stationen und Elemente der (mitunter weit zurück reichenden) geschichtlichen Entwicklung und der Versionshistorie berücksichtigt. Entwicklungspfade über mehrere gleichartige digitale Artefakte hinweg machen die Verschränkung von Struktur und Funktion sichtbar. Auf diese Weise kann deutlich werden, wie aufbauend auf vorhandenen Strukturen Veränderungen an diesen vorgenommen werden – und dass Strukturen zur Entdeckung neuartiger Nutzungsmöglichkeiten führen können, an die die ursprünglichen Entwickler nicht gedacht haben. So wird das Ineinander greifen von *Nutzen und Gestalten* deutlich (vgl. [Sc08a], [SK07]).

Bei der Textverarbeitung beispielsweise beschreibt der Unterschied von „digitalem“ und „analogem“ Text den entscheidenden Entwicklungsschritt. Er verspricht den größten Erkenntnisgewinn und Nutzen für die Lernenden. Die Analyse von Selbstbild, Weltbild und Verhalten wird ergeben (siehe den folgenden Abschnitt), dass Textverarbeitung als bekannt und daher wenig spannend eingeschätzt wird, bei gleichzeitig eher oberflächlicher Nutzung. Dieses scheinbare Paradox erklärt sich mit Hilfe der dualen Natur der Textverarbeitung: Während sie aussieht wie eine weiterentwickelte Schreibmaschine (Funktion), arbeitet sie dennoch nach fundamental anderen Prinzipien (Struktur), die aber nicht offensichtlich sind. Dieses Missverständnis führt dazu, dass Textverarbeitung als bessere Schreibmaschine und somit nur oberflächlich genutzt und wahrgenommen wird. Die Schreibmaschine arbeitet mit „analogem“ Text, das heißt mit dauerhaften visuellen Markierungen auf einem festen Beschreibstoff. „Digitaler“ Text zeichnet sich demgegenüber durch die Trennung von Code und Font aus. So wird nicht das visuelle Arrangement der Fonts, sondern der (Uni-)Code der Zeichen, zusammen mit Layoutanweisungen, gespeichert. Beim Schreiben werden Codes eingegeben, die von der Textverarbeitung in einer für das menschliche Auge nicht wahrnehmbaren Verzögerung in das visuelle Pendant umgerechnet werden. Dabei wird nicht nur der visuelle Teil des Zeichens berechnet und dargestellt, sondern auch der notwendige Absatz- und Seitenumbruch sowie eventuelle Umbrüche eingebetteter Objekte. Das digitale Artefakt Textverarbeitung vermittelt demgegenüber jedoch die Illusion, der Benutzer würde gleichsam auf virtuellem Papier schreiben und entsprechend den Text direkt formatieren ([Sc08a], [Sc08b], [Ha06]).

3. Rekonstruktion von Lernvoraussetzungen, Finden von Zielen

Mittlerweile verfügen Schülerinnen und Schüler (fast ausnahmslos) über vielfältige Erfahrungen mit unterschiedlichen digitalen Artefakten, bevor sie das erste Mal in der Schule eine Form informatischer Bildung erleben (sei das nun ITG, die Computer-AG oder der Informatikunterricht).

Verwunderlich ist jedoch, dass trotz allgemein umfangreicher Nutzungserfahrungen mit unterschiedlichen digitalen Artefakten diametral entgegengesetzte Computerbiografien entstehen (vgl. [SK07]): Für eine Gruppe bilden diese Erfahrungen die Eintrittskarte in die Welt der Informatik, für eine andere Gruppe bilden scheinbar fast identische Erfahrungen eine unüberwindbare Barriere, die den Einstieg in die Informatik verhindert. Die

einen sehen nur den Aspekt des Benutzens, sie fühlen sich zumeist als Outsider in Bezug auf Informatik und den professionellen Umgang mit digitalen Artefakten. Die anderen sehen auch Aspekte des Gestaltens, sie fühlen sich als Insider und schätzen sich als professionelle Benutzer ein. Nicht nur die Weltbilder unterscheiden sich, sondern auch die Selbstbilder und Handlungsmuster. Irgendwo müssen sich Qualität und subjektive Aufarbeitung oder Möglichkeit der Aufarbeitung ähnlich fundamental unterscheiden.

Die Suche nach der Antwort führt zum Ansatz der Dualität digitaler Artefakte. Damit kann der Unterschied des Erlebens auf die Dualität von Struktur und Funktion zurückgeführt und erklärt werden. In der Nutzung digitaler Artefakte wird zunächst die Funktion wahrgenommen und dementsprechend Funktionalität des Artefakts genutzt – in problematischen Nutzungssituationen und in der Nutzung komplexer Funktionalität muss zusätzlich auf erworbenes Strukturwissen über die innere Funktionsweise des Artefakts zurückgegriffen werden. Und eben dieses ist wohl nicht allen aufgrund von Benutzungserfahrungen zugänglich. Wenn sich jemand trotz intensiver Computernutzungserfahrungen als reiner Benutzer einschätzt, der mit Administration, dem Lösen von Benutzungsproblemen etc. (=dem professionellen Nutzen) überfordert ist, dann sind die eigenen Erfahrungen der Computernutzung entweder nur auf der Ebene der Funktion verarbeitet worden bzw. die in der Nutzung erworbenen Vorstellungen über die Struktur sind zu lückenhaft oder falsch. Jemand, der sich als Gestalter einschätzt, hat dagegen in viel höherem Ausmaß angemessene Vorstellungen der Struktur aufbauen können, die zum einem professionelle Nutzung ermöglichen und damit das bewusste Umgestalten der angebotenen Strukturen erlebbar machen. Von dieser Warte aus kann dann auch – im Kontrast zu den Nutzern – Softwareentwicklung und Programmieren als interessant, sinnvoll und nützlich erlebt werden.

Zum andern erklärt die Dualität, weshalb die professionelle Benutzung und die professionellen Benutzer (die ‚Informatiker‘) als unzugänglich, merkwürdig oder magisch beschrieben und erlebt werden. Ein ‚Outsider‘ kann sich schlicht nicht erklären, wieso ‚auf einmal‘ dann doch die im Text eingebettete Grafik an der richtigen Stelle bleibt. Denn auch der professionelle Benutzer nutzt dieselben Werkzeuge, ruft nur Funktionalität auf, aber bei ihm macht der Computer das, was er soll – versucht es aber der Benutzer alleine, dann scheint es, als höre der Computer nicht auf ihn. Wer nur die Seite der Funktion wahrnimmt, der kann eben nicht unter Rückgriff auf die Struktur die Situation erklären und zielgerichtet Funktionalität benutzen.

In *Computernutzungsbiographien*, den biographischen Studien zur Genese von lernrelevanten Einstellungen und Verhaltensweisen gegenüber der Informatik und dem Benutzen von verschiedenen digitalen Artefakten hat sich die Analyse von drei miteinander verzahnten Dimensionen als sehr ertragreich herausgestellt ([KS07], [SK07]). Diese drei Dimensionen werden daher auch für die Dualitätsrekonstruktion herangezogen. Die obige Diskussion der Diskrepanz zwischen der Wahrnehmung der Textverarbeitung als bessere Schreibmaschine und der tatsächlichen „Funktionsprinzipien“ zeigt, wie die Dualität von Struktur und Funktion hilft, relevante Unterschiede in Welt- und Selbstbildern sowie zugeordneten Handlungsmustern aufzudecken. Die Rekonstruktion von Welt- und Selbstbildern und Handlungsmustern spezifisch für ein digitales Artefakt erlaubt, spezifische

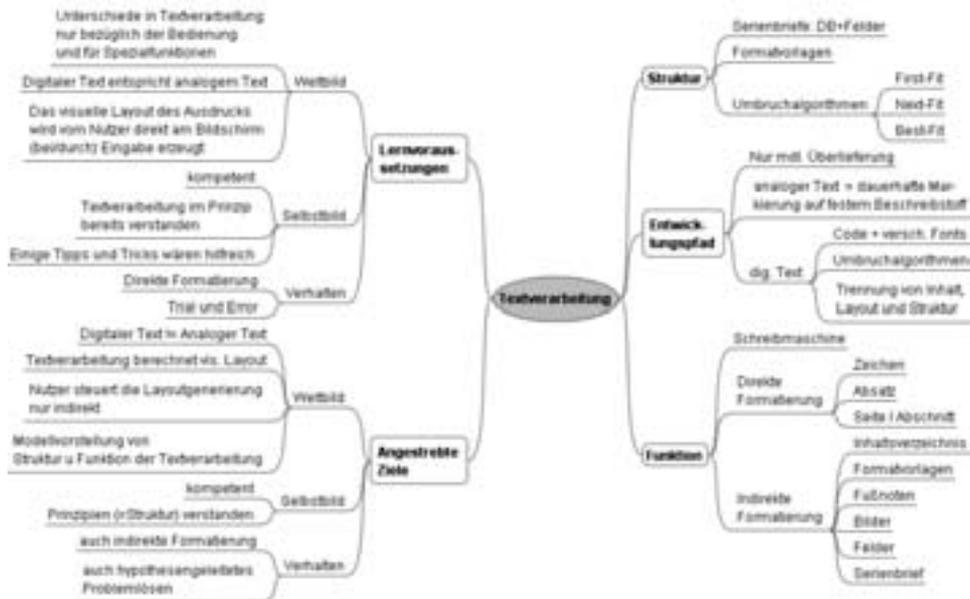


Abbildung 4: Didaktische Rekonstruktion der Textverarbeitung

Eine genauere Durchsicht der in der Mindmap gesammelten Inhalte und ihrer Sequenzierung wird weitere innere Vernetzungen aufzeigen: Digitaler Text als Grundlage für die notwendigen Umbruchalgorithmen, die wiederum Ausgangspunkt für weitere Layoutfunktionen sind und damit die indirekte Formatierung ermöglichen (vgl. Diskussion der Ideengeschichte in Abschnitt 2.). Diese aufbauenden Funktionen sind zumeist komplexe Funktionen (zu komplexen Funktionen vgl. „implicit operations“ und „explicit operations“ in [Sc08b]).

Generelles spezifisches Lernziel der Dualitätsrekonstruktion ist, ein angemessenes Bild des digitalen Artefakts aufzuzeigen, indem wesentliche Strukturelemente auf typische Funktionselemente bezogen werden und damit Grundlagen für kompetente Nutzungsstrategien der Schülerinnen und Schüler legen. Diese wiederum führen zu der unmittelbar erlebten Erfahrung, dass informatische Bildung im Alltag sinnvoll und nützlich sein kann.

Insgesamt sind verschiedene der oben rekonstruierten Lernvoraussetzungen, Ziele und Lerninhalte generisch; d.h. sie können auf andere Artefaktklassen angepasst und übertragen werden: relevantes Strukturwissen für kompetente Interaktion, hypothesengesteuertes Problemlösen mit Hilfe von Strukturwissen, angemessene Modellvorstellung des Zusammenhangs von Funktion und Struktur.

Die Darstellung des Schemas in Form einer Mindmap soll auf zweierlei Aspekte hinweisen: Zum einen geht es darum, in einem Ideensammelprozess wesentliche Gedanken kurz festzuhalten – was durchaus einschließen kann, dass zwischendurch Fragen auftreten, die recherchiert werden müssen. Zum anderen geht es darum, die Beziehungen zwischen den Hauptästen zu sehen: Zwischen Lernvoraussetzungen, Zielen und Eigenschaf-

ten des digitalen Artefakts. Das heißt beispielsweise, dass durch das Nachdenken über die Struktur ggf. deutlich wird, welche möglichen unzureichenden Präkonzepte bestehen.

In der folgenden Tabelle zeigt Leitfragen für die Dualitätsrekonstruktion:

Lern- vor- aus- set- zun- gen	Weltbild	Welche vorunterrichtlichen Vorstellungen über Funktion und Struktur des digitalen Artefakts könnten existieren? Welche Vorstellungen über typische Verwendungen und typische Verwender liegen vor?
	Selbstbild	Welche Einstellungen gegenüber dem dA werden vermutet? Z.B.: Interesse, Abwehr, eigene Einschätzung als kompetent oder nicht kompetent. Sieht man sich selbst als typischer oder möglicher Anwender?
	Handlungs- muster	Gibt es typische Nutzungsweisen der Lernenden? Welche Funktionen werden typischerweise benutzt? Welche (komplexen) Funktionen eher nicht? Gibt es typische Strategien im Umgang mit Fehlern oder komplexen Anwendungsfällen?
Ziele	Weltbild	Können wesentliche Struktur- und Funktionselemente erklären und aufeinander beziehen. Können den Zusammenhang von wesentlichen Struktur- und Funktionselementen nutzen, um die „Natur“ des digitalen Artefakts zu erklären. Verstehen, wie welche Strukturen beim Anwenden erzeugt werden, bzw. als Voraussetzung für die Nutzung komplexer Funktionen erzeugt werden müssen.
	Selbstbild	Können die eigene Rolle beim Benutzen reflektieren, haben das Selbstvertrauen, sich weiter einzuarbeiten (sowohl in Strukturen als auch in komplexe Funktionen)
	Handlungs- muster	Können Kenntnisse über Strukturen, Funktionen und komplexe Funktionen nutzen, um sich in neue Anwendungsszenarien einzuarbeiten. Können zielgerichtet mit dem digitalen Artefakts experimentieren, um die eigenen Vorstellungen über Struktur und Funktion zu prüfen. Können hypothesengeleitet Probleme in Anwendungsszenarien lösen, können zielgerichtet Ursachen für Defekte überlegen und prüfen, indem sie dazu auf die Verknüpfung von Struktur und Funktion zurückgreifen.
Digi- tales Arte- fakt	Funktion	Welches sind die wesentlichen Funktionen des digitalen Artefakts? Welches sind die wesentlichen komplexen Funktionen?
	Entwick- lungspfad	Wie haben sich typische Funktionen und Strukturen aus den Vorläufern (älteren Versionen, verwandten Artefaktklassen) entwickelt? Welche Erweiterungen / Veränderungen sind in naher Zukunft absehbar (z.B. angekündigte neue Versionen)? Welches sind die tragenden, d.h. erklärenden und sinngebenden Ideen dieser Entwicklung (aus Sicht der Funktion wie der Struktur)? Gibt es relevante 'analoge' Vorläufer des digitalen Artefakts? Welche spezifische Eigenschaften können durch Verweis auf diese Vorläufer erklärt werden?
	Struktur	Welches sind die wesentlichen Strukturen des digitalen Artefakts? Strukturen können sein: wichtige informatische Konzepte (Backtracking, Schichten, Geheimnisprinzip, Benutzungsmetapher); das Konzept der verwendeten Datenstruktur (Container als Listen, Graphen oder Bäume), (wesentliche Ideen der zentralen) Algorithmen.

Abbildung 5: Leitfragen für die Dualitäts-Rekonstruktion

Zudem sollten die folgenden Regeln der Dualitätsrekonstruktion berücksichtigt werden:

- **Exemplarische Inhalte herausarbeiten:** Es geht darum, die wesentlichen Aspekte herauszuarbeiten – eine vollständige Beschreibung von Funktions- und Strukturelementen eines digitalen Artefaktwürde jede Darstellung sprengen. Die Mindmap stellt also die für den Unterricht ausgewählten bzw. in Betracht kommenden Themen dar.

- **Innere Vernetzungen herausarbeiten:** Die einzelnen Aspekte sollen aufeinander bezogen werden. Wie im Beispiel gibt es immer logische und inhaltliche Beziehungen zwischen den Hauptästen. Angenommene Weltbilder in den Lernvoraussetzungen beziehen sich oft auf vermutete Defizite im Strukturwissen, daraus leiten sich Ziele ab. Zudem sollten die Funktionsaspekte deutlich werden, an denen Strukturelemente erklärt und als sinnvoll erlebt werden können.
- **Ideengeschichte herausarbeiten:** Im Entwicklungspfad, der Ideengeschichte des Artefakts, können längerfristig gültige Struktur- und Funktionsaspekte, oder gerade Brüche in diesen Aspekten deutlich werden. Diese Themen werden dann im Unterricht in Bezug auf ‚historische Einordnung‘ vermittelt – sie brauchen nicht eigens nochmal in Struktur oder Funktionsast aufgeführt werden (vgl. Abbildung 4). Das Verfahren sieht daher vor, zumindest kurz auf die Entwicklungsgeschichte einzugehen. Hintergrund ist die These, dass jedes aktuelle digitale Artefakt Merkmale aufweist, die nur vor einem solchen Hintergrund erklärbar sind. Nur durch Kenntnis der Entwicklungsgeschichte ist Verständnis für den Artefaktcharakter (das heißt das menschengemachte, das intentional gestaltete und veränderbare) erreichbar. Nur so wird die Rolle der Informatik deutlich. Das ist die Voraussetzung, um ein angemessenes Bild der Wissenschaft Informatik und ihrer Anwendungen zu vermitteln.
- **Aus Fakten Erkenntnisse gewinnen:** Die Ziele sollen stets auf allen drei Dimensionen liegen, um ein reines Vermitteln von Faktenwissen zu vermeiden. Im Sinne eines weiten Kompetenzbegriffs umfasst Lernen mehr als rein kognitive Aspekte, mehr als die Vermittlung von Faktenwissen. Neben motivationalen und willensmäßigen Aspekten sind auch handlungsrelevante Dimensionen zu berücksichtigen, die aus Fakten Erkenntnisse werden lassen. Erkenntnisse wirken handlungsleitend, geben Orientierung, werden zur Gewinnung weiteren Wissens eingesetzt, verknüpfen Fakten. Handlungsrelevanz muss mehr bedeuten als eine Operationalisierung zur Überprüfung des Faktenwissens. Der Begriff des Orientierungswissens mag diesen Aspekt in Abgrenzung zum Faktenwissen beschreiben.

5. Didaktisch-Methodische Konsequenzen und Schlussfolgerungen

5.1. Methodische Konsequenzen

Das Experimentieren zeichnet sich als eine geeignete Unterrichtsmethodik ab, um a) das hypothesengeleitete Problemlösen einzuüben, b) eine Brücke von der Funktion zur Struktur zu schlagen und c) so eine angemessene Modellvorstellung über den Zusammenhang von Funktion und Struktur zu entwickeln.

Die Lernenden experimentieren hypothesengeleitet mit einzelnen Funktionen und entdecken so die zu deren Erbringung implementierten Strukturen. Das Verfahren liefert eine Grundlage für entdeckenden Unterricht, den ich als *Experimenteller Zugang zu digitalen Artefakten* beschreiben möchte: Schülerinnen und Schüler entdecken den Zusammen-

hang von Funktion und Struktur; verbalisieren ihre Vermutungen über die Struktur; wenden Funktionalität gezielt an, d.h. bauen (Daten-) Strukturen auf; rufen einen Befehl auf und prüfen das Ergebnis in Bezug auf die Vorhersage, um so die Vermutung über Strukturelemente zu prüfen. Gleichzeitig lernen sie so Aspekte des professionellen Benutzens. Es entsteht eine *Brücke* für die Benutzer, auch professionelles Benutzen und Gestalten erleben zu können (vgl. [SK07]). Zudem: Wir können gezielt und systematisch diese Experimente entwickeln. Die bisherigen Erfahrungen zeigen, dass bei der Dualitätsrekonstruktion automatisch entsprechende Ideen entstehen. Beispielsweise könnte nach der Erarbeitung einiger unterschiedlicher Umbruchalgorithmen ein Experiment entwickelt werden, mit dem geprüft werden kann, welcher Algorithmus für den Umbruch am Bildschirm benutzt wird. Aufbauend darauf könnte mit komplexeren Layoutsituationen experimentiert werden, beispielsweise der Anordnung eingebetteter Grafiken (hier kann der Benutzer durch verschiedene Verankerungen der Grafik im Text unterschiedliche Layoutstrategien auswählen). An diesem Beispiel kann deutlich werden, dass das auf den ersten Blick für den Benutzer verwirrendes Verhalten („Wieso hat die verflixte Software schon wieder das Bild verschoben?“) durch Strukturwissen erklärt werden kann und wie nun durch stärker hypothesengeleitetes Vorgehen Benutzungsprobleme gelöst werden können.

5.2. Didaktische und bildungstheoretische Konsequenzen

Die duale Natur digitaler Artefakte bewirkt verschiedene Schwerpunktverlagerungen in der Betrachtung des Lerngegenstands: Die vermeintlich 'reine' Informatik als von der Struktur her kommend wird zunehmend Funktion einbeziehen – das kann im Informatikunterricht geschehen, indem z.B. Algorithmen als Struktur Aspekte digitaler Artefakte behandelt werden, anstelle losgelöster Unterrichtseinheiten zu Algorithmen. Umgekehrt muss 'reine' Anwenderschulung, von der Funktion her kommend, zunehmend die Struktur einbeziehen. Damit sind folgende Schwerpunkte in den Richtzielen informatischer Bildung verknüpft (vgl. [Sc08b]): A) Vermitteln von Orientierungswissen, bzw. Aufklärung über die Natur der digitalen Artefakte, die den Einzelnen umgeben und dessen Alltag prägen. B) Vermitteln von angemessenen Nutzungskompetenzen, die über Anwenden einfacher Funktion hinausgehen und komplexe Funktionen, das Anpassen vorhandener Strukturen und ggf. die Erweiterung um neue Strukturelemente einschließen. C) Vermitteln von Orientierungswissen über Verfahren, Werkzeuge und Entwicklungslinien der Wissenschaft Informatik, die die Entwicklung digitaler Artefakte ermöglicht und die (zum Teil weitreichende) Ideengeschichte verstehbar macht.

6. Diskussion der wesentlichen Ziele des Verfahren

Das vorgestellte Verfahren hat drei wesentliche Ziele (vgl. den Anfang des Artikels): Aktuelle Gegenstände und Beispiele für den Informatikunterricht zugänglich machen; schülerorientierte Unterrichtsplanung und -gestaltung, Verdeutlichung des Alltagsbezugs informatischer Bildung. Im Folgenden werden diese drei Anforderungen aus unterrichtspraktischer Sicht und im Hinblick auf fachdidaktische Forschungsfragen diskutiert.

A) Aktualität ermöglichen

Aktualität kann natürlich auf vielfältige Art und Weise hergestellt werden (Aufgreifen von Medienberichten, aktuelle Schlagzeilen aus der Forschung, Ereignisse an der Schule, aktuelle Anforderungen aus anderen Unterrichtsfächern, etc.). Hier wird (nur) ein Weg vorgeschlagen: Informatische Inhalte anhand der Verzahnung von Funktion und Struktur aktueller digitaler Artefakte zu verdeutlichen. Dieser Weg wird insbesondere deshalb vorgeschlagen, um – schlagwortartig formuliert – ITG und Informatik zu verknüpfen, die sich klassischerweise entweder fast ausschließlich auf die Dimension der Funktion oder fast ausschließlich auf die Dimension der Struktur beziehen. Das Verfahren erleichtert es, jeweils neue und neuartige digitale Artefakte mit dem soeben formulierten Anspruch zielgerichtet für den Unterricht aufzuarbeiten. Eine gründliche Dualitäts-Rekonstruktion sprengt jedoch bei weitem den Aufwand für die alltägliche Unterrichtsvorbereitung. Allerdings kann das Verfahren entsprechende fachdidaktische Forschungsprojekte anregen.

Zudem ergibt sich auf mittlere Sicht ein deutlicher Vorteil, wenn man sich die Bedeutung des Entwicklungspfad ansieht: Auch neue, d.h. stark veränderte Artefakte weisen in ihrem Entwicklungspfad wesentliche Inhalte auf, die durch eine neue Version nicht entwertet werden – Analoges gilt für neuartige Artefakte, die zumeist aus Erfahrungen und Ideen mit anderen Artefakten und den dort verwendeten Strukturen und Funktionen entstehen. Positiv formuliert: Hat eine Lehrkraft sich auf diese Weise eine Artefaktklasse erschlossen, dann können wesentliche Anteile auf neue Artefakte dieser Klasse – und damit wesentliches didaktisch-methodisches Wissen – übertragen werden. In diesem Sinne erleichtert das Verfahren Aktualität.

B) Schülerorientierte Unterrichtsplanung

Auch Schülerorientierung wird durch vielfältige Maßnahmen erreicht, eine wesentliche Bedingung für Schülerorientierung liegt beispielsweise in der konkreten Unterrichtsdurchführung.

Für die Forschung eröffnet sich hier ein Feld, das schlagwortartig mit dem Begriff der Artefaktbiographie bezeichnet werden kann. Artefaktbiographien erheben spezifische auf eine Artefaktklasse bezogene Lernvoraussetzungen, einschließlich möglicher Lernhürden im Sinne negativer Attribuierungen oder Lernhemmungen (die auch im übersteigerten Selbstbewusstsein liegen können). Damit werden Lernverläufe, Lernbiographien und spezifische Alltagsvorstellungen über bestimmte Artefakte erhoben und können für einen schülerorientierten Unterricht genutzt werden. Zur Zeit liegen solche Forschungen, wenn überhaupt, nur vereinzelt vor. Aber auch in der praktischen Planung sollte die gedankliche Rekonstruktion der Lernvoraussetzungen die Schülerorientierung erhöhen. Zudem erlaubt der Ansatz die spezifische Diskussion und Auswahl von Lerninhalten und Lernzielen (viel genauer als: Die SuS lernen wesentliche Konzepte der Textverarbeitung kennen, nämlich: Auflistung wesentlicher Konzepte, Abgrenzung zu anderen Artefaktklassen, Kombination mit typischen Anwendungsszenarien). Schülerorientierung in Bezug auf die Unterrichtsdurchführung wird zudem ein wenig in Bezug auf möglichen entdeckenden Unterricht mit Hilfe von Experimenten gefördert (vgl. Abschnitt 5.1.).

C) *Den Alltagsbezug informatischer Bildung verdeutlichen*

Konkrete Gegenstände des Unterrichts (die behandelten digitalen Artefakte) verdeutlichen die Relevanz informatischer Bildung. Durch entsprechende Auswahl zu thematisierender digitaler Artefakte entsteht ein nachvollziehbarer Alltagsbezug für die Schülerinnen und Schüler. Wesentlicher Punkt ist das Erlebnis, wie informatische Kenntnisse die Funktionsweise digitaler Artefakte erklären und so helfen, komplexe Nutzungsszenarien zu bewältigen. Diese Relevanz im Alltag kann immer wieder neu an neuen Artefakten und dem Aufdecken ihrer Strukturen erlebt werden. Zudem werden auf diese Weise abstrakte, theoretische Konzepte der Informatik in konkreten Anwendungssituationen, in sinnstiftenden Kontexten bzw. alltäglichen oder zumindest zugänglichen Nutzungssituationen verankert.

7. Literaturverzeichnis

- [Ha06] Haigh, Thomas: Remembering the Office of the Future: The Origins of Word Processing and Office Automation. In: IEEE Annals of the History of Computing, October-December 2006. S. 6-31.: <http://tomandmaria.com/tom/Writing/Annals2006WP.pdf>
- [KS07] Knobelsdorf, Maria; Schulte, Carsten: Das informatische Weltbild von Studierenden. In: Schubert, Sigrid (Hg.): Didaktik der Informatik in Theorie und Praxis. 12. GI-Fachtagung Informatik und Schule - INFOS 2007, Siegen, S. 69–79.
- [Sc08a] Schulte, Carsten: Duality Reconstruction – Teaching Digital Artifacts from a Socio-technical Perspective. In: Informatics Education - Supporting Computational Thinking. Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2008. S. 110-121
- [Sc08b] Schulte, Carsten: Die duale Natur digitaler Artefakte als Kern Informatischer Bildung. In: Münsteraner Workshop zur Schulinformatik 2008. S. 7-24
- [SK07] Schulte, Carsten; Knobelsdorf, Maria: Attitudes towards computer science - computing experiences as a starting point and barrier to computer science. In: Proceedings of the third international workshop on Computing education research, ICER '07, S. 27-38, 2007, <http://doi.acm.org/10.1145/1288580.1288588>.

Referenzprojekte für den Informatikunterricht

Unterrichtsprojekte zwischen Selbst- und Fremdbestimmung

Jochen Koubek

Humboldt-Universität zu Berlin

Institut für Informatik

Informatik in Bildung und Gesellschaft

jochen.koubek@hu-berlin.de

Abstract: Der Kern des IT-Weiterbildungssystems sind die Referenzprojekte der APO-IT (Arbeitsprozessorientierten IT-Weiterbildung). Trotz ihrer Ausrichtung auf arbeitsbegleitende Prozesse von Berufstätigen lassen sich aus ihr zahlreiche Anregungen für den Schulunterricht und die Entwicklung zeitgemäßer Curricula ableiten. Referenzprojekte sind das Bindeglied zwischen dem selbstbestimmten und ergebnisorientierten Unterricht der Bildungsstandards und dem fremdbestimmten und inhaltsorientierten Unterricht der Lehrpläne und ausgearbeiteter Unterrichtsentwürfe. Als Patterns für Unterrichtsentwürfe sind sie eine qualitative Weiterentwicklung der Bildungsstandards.

Der vollständige Beitrag ist erschienen in:

Peters, I.-R. (Hrsg.): Informatische Bildung in Theorie und Praxis – 25 Jahre »INFOS – Informatik und Schule«. INFOS 2009 – 13. GI-Fachtagung Informatik und Schule, 21. - 24. September 2009 in Berlin. LOG IN Verlag, Berlin, 2009.

Autorenverzeichnis

B	Humbert, Ludger 353
Berndt, Elin-Birgit 230	J
Bischof, Ernestine 50	Jurević, Diana 316
Braxmeier, Philippe 304	K
Brinda, Torsten 157	Kiesmüller, Ulrich 169
Büttner, Katrin 62	Knapp, Thomas 62
D	Knobelsdorf, Maria 218
Döbeli Honegger, Beat 304	Koerber, Bernhard 3
Dohmen, Michael 329	Kohl, Lutz 328
E	Kollee, Christian 133
Ehlert, Albrecht 121	Koubek, Jochen 268, 367
Engbring, Dieter 329	M
F	Magenheim, Johannes 329
Fothe, Michael 330	Micheuz, Peter 243
Freudenberg, Rita 86	Mittermeir, Roland T. 50
Frey, André 304	Müller, Jürgen 340
Friedrich, Steffen 62	O
G	Oldenburg, Reinhard 146
Gallenbacher, Jens 28, 257	P
H	Pasternak, Arno 280
Hartmann, Werner 316	Pohl, Wolfgang 38
Hein, Hans-Werner 38	R
Heming, Matthias 134	Rabel, Magnus 146
Herper, Henry 74	Roland, Holger 293
Hielscher, Michael 281	
Hinz, Volkmar 74	

Romeike, Ralf 255

S

Sack, Lothar 205
Schauer, Hanno 354
Schlüter, Kirsten 38, 181
Schubert, Sigrid 13, 133
Schulte, Carsten 121, 268, 355
Schulze, Peter 268
Schwidrowski, Kirstin 73
Spalteholz, Wolf 193
Stechert, Peer 133
Stöcklin, Nando 316

V

Vahrenhold, Jan 280
van de Water, David 157

W

Wach, Christian 256
Wagenknecht, Christian 281
Weber, Markus 109
Weigend, Michael 97
Wiesner, Bernhard 109
Witten, Helmut 205, 268

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. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelrath, 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 Magenheimer (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 Magenheimer, 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 Brömmme, Christoph Busch (Eds.): BIOSIG 2003: Biometric 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 Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (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 Ranneberg, 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, Wolfried 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 Brey (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 Röbling, 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 Walther (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 Gimmich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)
- P-127 Thomas Kühne, Wolfgang Reising, 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 and 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 Hühnlein (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-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und
Schule«

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

