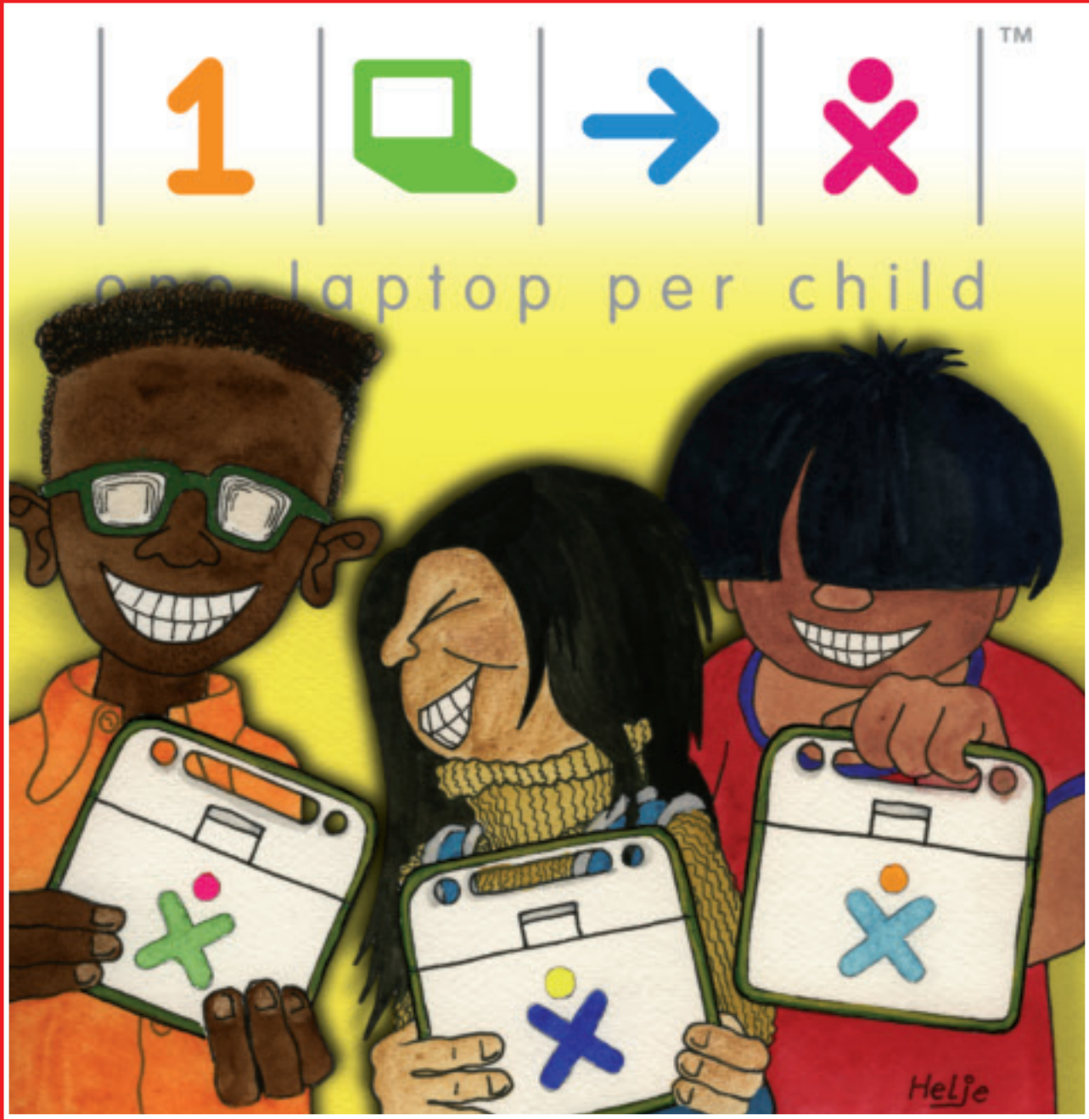
www.log-in-verlag.de/

Mit XO-LiveCD
One Laptop per Child

A 12501 F

LOG IN

Informatische Bildung und Computer in der Schule



Ein Laptop für jedes Kind (OLPC). Lernen mit Etoys. Gestaltendes Lernen. SUGAR.

Programmieren mit dem XO-Laptop. Sprechende Katze und Zeichenschildkröte. Die XO-LiveCD.

Nr. 156
2009
LOG IN Verlag

Suchen oder vermissen Sie ein Themenheft von LOG IN?

Der LOG IN Verlag bietet Ihnen die Möglichkeit, aus Restbeständen einzelne LOG-IN-Hefte von 1993 bis 2003 verbilligt zu erstehen. Wählen Sie unter den noch lieferbaren Titeln.

▷ Einzelheft: 4,00 Euro
▷ Doppelheft: 8,00 Euro

(zusätzlich Versandkosten – bei Bestellungen ab 40,- Euro versandkostenfrei). Senden Sie bitte Ihre Bestellung per Fax an die Redaktion:

▷ **(030) 83 85 67 22**

oder per Post an den Verlag:

▷ **LOG IN Verlag GmbH**
Postfach 33 07 09
14177 Berlin

Die Inhaltsangaben der einzelnen Hefte finden Sie im Internet (http://www.log-in-verlag.de/) im „LOG IN archiv“ und dieses Bestellformular als PDF-Datei zum Ausdrucken unter „Raritäten“.

☐ … Expl. Nr. 131/132, 2004:
Komponentenbasierte Projektentwicklung – u. a.: Komponentenbasierte Softwareentwicklung. Suchbaum-Modellierung. Rekonstruktives Modellieren.

☐ … Expl. Nr. 130, 2004:
Künstliches Leben – u. a.: Künstliches Leben – ein Überblick. Leben in der rekursiven Welt. Virtuelle Ameisenwelt.

☐ … Expl. Nr. 128/129, 2004:
Objektorientiertes Modellieren und Programmieren – u. a.: Möglichkeiten und Grenzen maschineller Intelligenz. Suchverfahren zur Problemlösung.

■ Nr. 127, 2004 *(nur geringer Restbestand):*
Ergonomische Rechnerräume – u. a.: Ergonomie am PC. Pädagogische Ergonomie. Grafik im Anfangsunterricht.

☐ … Expl. Nr. 126, 2003:
Digitale Klangwelten – u. a.: Digitale akustische Signale. Klänge sehen – Funktionen hören. Musik mit JAVA.

☐ … Expl. Nr. 125, 2003:
Mobiles Rechnen – u. a.: Allgegenwärtiges Rechnen. Elektronisches Lernen mit Taschencomputern. Genetische Algorithmen.

■ Nr. 124, 2003 *(nur geringer Restbestand):*
Informatische Bildung: Sekundarstufe II – u. a.: Das Problem der speisenden Philosophen. Wann sind zwei Objekte gleich?

☐ … Expl. Nr. 122/123, 2003:
Informatische Bildung: Sekundarstufe I – u. a.: Lehrerbildung und Informatik. Von der rezeptiven zur konstruktiven Internetnutzung. E-Mail für Einsteiger.

☐ … Expl. Nr. 121, 2003:
Informatische Bildung: Primarbereich – u. a.: Grundschule in der Wissensgesellschaft. Zählen Computerspiele zur Kultur?

☐ … Expl. Nr. 120, 2002:
Lernen mit elektronischen Medien – u. a.: Internetrecherche im Informatikunterricht. Interaktive Animationen.

☐ … Expl. Heft 5/6, 2001:
Digitale Bilderwelten – u. a.: Computergrafik. Geometrisches Modellieren. Objektorientiertes Modellieren.

☐ … Expl. Heft 3/4, 2001:
Systemverwaltung – u. a.: Folgekosten der IT-Investitionsprogramme. Rechnerbetreuung in Schulen. Fernwartung.

☐ … Expl. Heft 2, 2001:
Programmieren in der informatischen Bildung – u. a.: Assoziieren und Spezialisieren. Programmverifikation.

☐ … Expl. Heft 1, 2001:
Informatik heute und morgen – u. a.: Zufall und zufallsgesteuerte Algorithmen. Informatik-Anfangsunterricht.

■ Heft 6, 2000 *(nur geringer Restbestand):*
Visionen der Informatik

☐ … Expl. Heft 5, 2000:
Medienkompetenz mit Computern – u. a.: Computerbasierte Medien. Webbasierte Lernprogramme.

☐ … Expl. Heft 3/4, 2000:
Intelligente Agenten – u. a.: Wettkampf der Prozesse. Schulische Rechnernetze.

☐ … Expl. Heft 2, 2000:
Neue IT-Berufe – u. a.: Curriculare Orientierungen. Das Rucksackproblem.

☐ … Expl. Heft 1, 2000:
Publizieren im Netz – u. a.: Vom Schriftsatz zum Web-Design.

■ Heft 6, 1999 *(nur geringer Restbestand):*
Moderne Medienwelten

☐ … Expl. Heft 5, 1999:
Recht und Informatik

☐ … Expl. Heft 3/4, 1999:
Telearbeit und Telekooperation

☐ … Expl. Heft 2, 1999:
Informatik und Philosophie

■ Heft 1, 1999 *(nur geringer Restbestand):*
Intranet – Aufbau und Nutzung in der Schule

☐ … Expl. Heft 6, 1998:
Virtuelle Realität

☐ … Expl. Heft 5, 1998:
Automatisierung

☐ … Expl. Heft 3/4, 1998:
Suchen und Finden im Internet

☐ … Expl. Heft 2, 1998:
Informatik und Mathematik

☐ … Expl. Heft 1, 1998:
Multimediale Autorensysteme

☐ … Expl. Heft 6, 1997:
Informatische Bildung und Internet

☐ … Expl. Heft 5, 1997:
Programmieren weltweit

☐ … Expl. Heft 3/4, 1997:
Programmiersysteme

☐ … Expl. Heft 2, 1997:
Lokale Netze in Schulen

■ Heft 1, 1997 *(nur geringer Restbestand):*
Multimedia in der Schule

■ Heft 5/6, 1996 *(nur geringer Restbestand):*
Kryptographie und Sicherheit in Netzen

■ Heft 4, 1996 *(nur geringer Restbestand):*
PCs und weltweite Netze als Arbeitshilfe für Lehrkräfte

■ Heft 3, 1996 *(vergriffen)*

■ Heft 2, 1996 *(nur geringer Restbestand):*
Computereinsatz in der Medizin

■ Heft 1, 1996 *(nur geringer Restbestand):*
Lehrerbildung

☐ … Expl. Heft 5/6, 1995:
Fuzzy-Logik

■ Heft 4, 1995 *(nur geringer Restbestand):*
Computer, Kreativität und Ästhetik

☐ … Expl. Heft 3, 1995:
Computereinsatz bei Behinderten

☐ … Expl. Heft 2, 1995:
Bildbearbeitung

☐ … Expl. Heft 1, 1995:
Anfangsunterricht.

■ Heft 5/6, 1994 *(nur geringer Restbestand):*
Datenfernübertragung und informatische Bildung

☐ … Expl. Heft 4, 1994:
Algorithmen und Datenstrukturen für den Unterricht

☐ … Expl. Heft 3, 1994:
EDV in der Landwirtschaft

☐ … Expl. Heft 2, 1994:
Datenbanken in der Schule

☐ … Expl. Heft 1, 1994:
Planung und Durchführung von Unterricht (Teil II)

■ Heft 6, 1993 *(vergriffen)*

☐ … Expl. Heft 5, 1993:
Parallelverarbeitung

■ Heft 4, 1993 *(vergriffen)*

☐ … Expl. Heft 3, 1993:
Datenfernübertragung für Schulen

■ Heft 1/2, 1993 *(nur geringer Restbestand):*
Multimedia im Unterricht

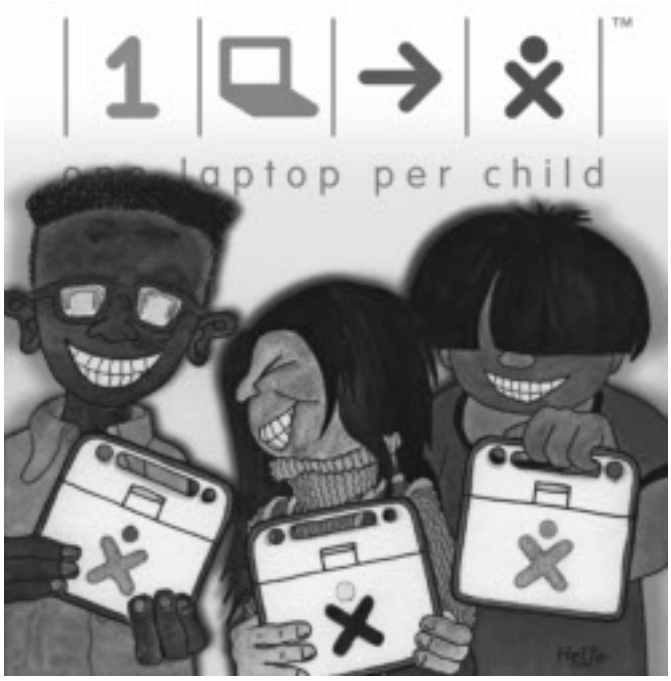
Absender:

Name, Vorname: _____ Datum: _____

Straße: _____

PLZ und Ort: _____ Unterschrift: _____

INHALT



ZUM THEMA

Ein Laptop für jedes Kind

Die Forderung, in Deutschland jede Schülerin und jeden Schüler mit einem Laptop für den Unterricht auszustatten, ist nicht neu, aber bislang nicht realisiert worden. Zwar wurden einzelne Klassen schon mit Laptops ausgestattet, und es wurden sogar Unterrichtskonzepte dafür entwickelt und erprobt, doch aus vielfältigen Gründen sind dies nur vereinzelte Versuche geblieben. Im Gegensatz dazu ist eine Initiative gestartet worden, deren Leitgedanke es ist, vor allem in sogenannten Schwellenländern und in der dritten Welt allen Kindern den Zugang zu aktuellem Wissen mit einem speziellen Laptop zu ermöglichen, der darüber hinaus zu einem kindgerechten und vielseitigen Lernwerkzeug umgestaltet wurde. Und so liegen aus dem Projekt *One Laptop per Child* (OLPC) mittlerweile erste Erfahrungen vor, die Anregungen bieten, sich auch hierzulande wieder mit diesem Thema auseinanderzusetzen.

Das Titelbild zum Thema wurde von Jens-Helge Dahmen, Berlin, für LOG IN gestaltet.

Impressum 2

Editorial 3

Berichte 4

THEMA

One Laptop per Child –
Von einer Vision zur globalen Initiative
von Christoph Derndorfer 12

Das OLPC-Pilotprojekt in Äthiopien
von Hermann Härtel 18

Lernen mit Etoys – weltweit
von Rita Freudenberg 23

Gestaltendes Lernen –
„Learning by design“ im Schulunterricht?
von Carmen Zahn 27

DISKUSSION

Pädagogische Konzepte versus Hardware –
Was können wir in Deutschland aus dem
OLPC-Projekt lernen?
Eine Diskussion zwischen Richard Heinen
und Joachim Wedekind 36

PRAXIS & METHODIK

SUGAR – ein Betriebssystem zum Lernen
von Rita Freudenberg 40

Programmieren mit dem XO-Laptop
von Joachim Wedekind und Christian Kohls 45

Sprechende Katze und Zeichenschildkröte –
Erste Schritte im visuellen Programmieren
mit *Turtle Art* und SCRATCH
von Rüdeger Baumann 51

Ein Ausflug in den Compilerbau – Syntaktische
Analyse und Berechnung arithmetischer Ausdrücke
von Torsten Brandes 59

COMPUTER & ANWENDUNGEN

Hardware:
Der XO-Laptop im Netz 65

Software:
Die XO-LiveCD 70

Hardware & Software:
Digitale Fenster öffnen –
Wie interaktive Tafeln Unterricht und Lernkultur
einer Schule verändern können (Teil 1) 73

FORUM

Rezension:
Kleuker, Stephan: Grundkurs Software-Engineering
mit UML – Der pragmatische Weg zu erfolgreichen
Softwareprojekten 77

Hinweise auf Zeitschriften 78

Computer-Knobelei:
Das Pomeranzenproblem 78

Veranstaltungskalender 79

Vorschau 80

LOG OUT 80

Beilage: XO-LiveCD – One Laptop per Child.

Herausgeber

Fachbereich Erziehungswissenschaft und Psychologie der Freien Universität Berlin,

zusammen mit

der **Gesellschaft für Informatik (GI) e. V.**, Bonn,
dem **Arbeitsbereich Prozesstechnik und berufliche Bildung** der Technischen Universität Hamburg-Harburg,
dem **Fachbereich Informatik** der Universität Dortmund,
dem **Fachbereich Informatik und Elektrotechnik** der Universität Siegen,
der **Fakultät Informatik** der Technischen Universität Dresden,
dem **Institut für Informatik** der Universität Stuttgart,
dem **Institut für Informatik** der Universität Zürich und
dem **Institut für Informatik-Systeme** der Alpen-Adria-Universität Klagenfurt.

LOG IN wurde 1981 als Fachzeitschrift aus den Informationsschriften „INFO – ein Informationsblatt zur Integration der Informatik in Berliner Schulen“ (1975–1979) des Instituts für Datenverarbeitung in den Unterrichtswissenschaften, Berlin, und „log in – Mitteilungen zur Informatik in der Schule“ (1979–1980) des Instituts für die Pädagogik der Naturwissenschaften, Kiel, begründet.

Redaktionsleitung

Bernhard Koerber (verantwortlich).

Freie Universität Berlin, FB Erziehungswissenschaft u. Psychologie
GEDiB – Redaktion LOG IN
Habelschwerdter Allee 45, D-14195 Berlin
Telefon: 030-83 85 63 36 – Telefax: 030-83 85 67 22
E-Mail: redaktionspost@log-in-verlag.de
URL: http://www.log-in-verlag.de/wwwredlogin/index.html

Bitte senden Sie Manuskripte für Beiträge, Anfragen zum LOG-IN-Service und sonstige Korrespondenz an die Redaktionsleitung.

Redaktion

Rüdeger Baumann, Garbsen; Jens-Helge Dahmen, Berlin (Grafik); Heinz Faatz, Berlin (Layout); Hannes Gutzer, Halle/Saale; Gabriele Kohse, Berlin (Redaktionssekretariat); Jürgen Müller, Gera; Ingo-Rüdiger Peters, Berlin (stellv. Redaktionsleitung); Achim Sahr, Berlin; Helmut Witten, Berlin.

Ständige Mitarbeit

Werner Arnhold, Berlin (Colleg); Günther Cyranek, Zürich (Berichte: Schweiz); Jens Fleischhut, Berlin (DV in Beruf & Alltag); Annemarie Hauf-Tulodziecki, Soest (Praxis & Methodik: Informatische Bildung in der Sekundarstufe I); Hanns-Wilhelm Heibey, Berlin (Datenschutz); Alfred Hermes, Jülich (Praxis & Methodik: Werkstatt); Ingmar Lehmann, Berlin (Praxis & Methodik: Informatik im Mathematikunterricht); Ernst Payerl, Erlensee (Praxis & Methodik: Informatische Bildung in der Sekundarstufe II); Sigrid Schubert, Siegen (Fachliche Grundlagen des Informatikunterrichts); Andreas Schwill, Potsdam (Aktuelles Lexikon); Joachim Wedekind, Tübingen (Praxis & Methodik: Informatik in naturwissenschaftlichen Fächern).

Verantwortlich für die Mitteilungen des Fachausschusses „Informatische Bildung in Schulen“ (FA IBS) der Gesellschaft für Informatik (GI) e. V. ist der Sprecher des Fachausschusses, Steffen Friedrich (Dresden).

Wissenschaftlicher Beirat

Wolfgang Arlt, Berlin; Peter Diepold, Göttingen; Steffen Friedrich, Dresden; Peter Gorny, Oldenburg; Rul Gunzenhäuser, Stuttgart; Immo O. Kerner, Nienhagen; Wolf Martin, Hamburg; Peter Micheuz, Klagenfurt; Helmut Schauer, Zürich; Sigrid Schubert, Siegen; Peter Widmayer, Zürich.

Mitarbeit an dieser Ausgabe

Torsten Brandes, Herbert Bruderer, Christoph Derndorfer, Ira Diethelm, Michael Fothe, Rita Freudenberg, Hermann Härtel, Richard Heinen, Volkmar Hinz, Thomas Iser, Christian Kohls, Wolfgang Rohrmoser, Carmen Zahn.

Koordination des Themenschwerpunkts in diesem Heft:
Joachim Wedekind.

Bezugsbedingungen

LOG IN erscheint fünfmal jährlich (4 Einzelhefte, 1 Doppelheft).

Abonnementpreis (4 Einzelhefte zu je 72 Seiten, 1 Doppelheft): Inland 59,80 EUR, Ausland 66,40 EUR, jeweils inkl. Versandspesen.
Ausbildungsabonnement: 20 % Ermäßigung des Abonnementpreises (nach Vorlage einer Studien- oder Referendariatsbescheinigung).
Einzelheft: 16,00 EUR, Doppelheft: 32,00 EUR, jeweils inkl. Versandspesen.
Die Preise enthalten bei Lieferung in EU-Staaten die Mehrwertsteuer, für das übrige Ausland sind sie Nettopreise.

Bestellungen nehmen der Verlag, die Redaktion oder jede Buchhandlung an. Die Kündigung von Abonnements ist mit einer Frist von 8 Wochen zum Ende jedes Kalenderjahres möglich.

Mitglieder der Gesellschaft für Informatik, die als Lehrer an allgemein- oder berufsbildenden Schulen oder als Dozenten tätig sind, können die Zeitschrift im Rahmen ihrer Mitgliedschaft beziehen.

Verlag

LOG IN Verlag GmbH
Postfach 33 07 09, D-14177 Berlin
Friedrichshaller Straße 41, D-14199 Berlin
Telefon: 0178 5 60 46 69 – Telefax: 030-8 62 16 45
E-Mail: abopost@log-in-verlag.de
URL: http://www.log-in-verlag.de/

Verantwortlich für den Anzeigenteil: Ingo-Rüdiger Peters,
Telefon: 030-83 85 63 36 (Anschrift siehe Redaktionsleitung).
Anzeigenverkauf: Hagen Döhner Media-Service,
Telefon: 0511-55 23 25 – Telefax: 0511-55 12 34.
Zurzeit gilt die Anzeigenpreisliste Nr. 24 vom 1. Januar 2009.

© 1993 LOG IN Verlag GmbH

Die Zeitschrift und alle in ihr enthaltenen Beiträge und Abbildungen sind urheberrechtlich geschützt. Mit Ausnahme der gesetzlich zugelassenen Fälle – insbesondere für Unterrichtszwecke – ist eine Verwertung ohne Einwilligung des Verlags strafbar.

Satz/DTP: FU Berlin – FB ErzWiss./Psych. – GEDiB, Berlin.
Belichtung und Druck:
MediaBogen Fiedler-Klotz-Nöschel GbR, Berlin.
Versand: DKS-Integral GmbH, Berlin.

LOG IN erscheint 2009 im 29. Jahrgang.

ISSN: 0720-8642

Laptops statt Schulranzen

Selten haben es ambitionierte Bildungsprojekte bis auf die Feuilletonseiten der etablierten Zeitungen und Zeitschriften geschafft. Dem Projekt *One Laptop per Child* (OLPC) ist das gelungen. Seit Nicholas Negroponte – Professor für Medientechnologie und Direktor des berühmten, von ihm mitgegründeten *MIT Media Laboratory* sowie Autor des Bestsellers „Being digital“ (deutsche Ausgabe: „Total digital“) – auf dem Wirtschaftsgipfel 2005 in Davos diese Initiative vorgestellt hat, ist sie dauerhaft in den Medien präsent. Nun mag das auch daran liegen, dass diese Initiative in der Öffentlichkeit vielfach auf den XO-Laptop reduziert wird, den sogenannten 100-Dollar-Laptop, der immerhin den Boom billiger Netbooks ausgelöst hat.

Leitgedanke ist – so wird es beim OLPC-Projekt formuliert –, Computer zu einer freien Wissensdatenbank und zu einem kindgerechten und vielseitigen Lernwerkzeug für die Schule umzugestalten und zusätzlich den Zugang zu aktuellem Wissen mithilfe digitalisierter Medien aller Art zu ermöglichen. Angestrebt wird die Verwendung Freier bzw. Open-Source-Software.

Natürlich ist der Ansatz der OLPC-Initiative nicht unumstritten geblieben. Vielfach wird die Frage gestellt, ob es nicht sinnvoller sei, dafür zu sorgen, dass die Kinder in den Entwicklungsländern erst einmal ausreichende Ernährung, sauberes Wasser und ein funktionierendes Schulsystem erhalten, bevor sie mit einem „High-Tech-Spielzeug“ ausgerüstet werden. Nach mehr als drei Jahren Laufzeit mehren sich auch die Stimmen mit der Frage, inwieweit das Projekt seinen ursprünglichen Zielen verhaftet geblieben ist oder durch Kooperation mit der Firma Microsoft einen grundlegenden Wandel vollzieht.

All dies wird in einem großen und aktiven Umfeld, das heißt auf vielen Wikiseiten und in vielen Weblogs intensiv diskutiert. Grundsätzlich hat das Projekt eine riesige Zahl von Unterstützern gefunden. Viele haben sich darangemacht, die Softwareausstattung des Rechners zu erweitern, sich an der Weiterentwicklung von Betriebssystem und Werkzeugen zu beteiligen sowie Unterrichtsmaterialien zu entwickeln.

Der Bekanntheitsgrad unter deutschen Lehrerinnen und Lehrern ist dennoch vergleichsweise gering. Das mag daran liegen, dass komplementäre Projekte an deutschen Schulen kaum möglich sind, da die XO-Laptops bisher nur an Regierungsorganisationen der Schwellenländer verkauft werden und allenfalls über ein Spendenprogramm (*Give Many Get Many*) erhältlich wären. Als gemeinnützige Gesellschaft organisiert ist der Träger des OLPC-Projekts nicht auf Gewinnmaximierung ausgerichtet. So beschränken sich bei uns die konkreten Erfahrungen mit dauerhafter individueller Verfügbarkeit von Rechnern und Zugang zum Internet auf die vereinzelten Versuche mit Notebook-Klassen (vgl. z.B. LOG IN 144/2007).

Vielleicht ist aber auch das explizite Lehr-/Lernmodell, auf das sich die OLPC-Initiative beruft, der Konstruktionismus, für Lehrerinnen und Lehrer an unseren Schulen eine Einstiegshürde, da sie in ein Netz curricularer Vorgaben und organisationaler Rahmenbedingungen eingebunden sind, was dessen konsequente Umsetzung nicht gerade begünstigt.

Andererseits kommen mehrere zentrale Aspekte des OLPC-Projekts aktuellen gesellschaftlichen Entwicklungen durchaus entgegen. Die hohe Durchdringung von beruflichem, aber inzwischen auch privatem Alltag mit Computern bzw. leistungsfähigen Mobilkommunikationsgeräten und dem damit gegebenen Zugang zum Internet (vgl. LOG IN 125/2006 und 145/2007) stellen auch Grundschulen hierzulande vor die Frage, wie dies medienpädagogisch, (fach-)didaktisch und unterrichtsmethodisch aufgegriffen und integriert werden kann.

Denn ein weiteres Ziel des OLPC-Projekts ist neben der Verbesserung der Schulausbildung grundsätzlich die Überwindung der sogenannten digitalen Spaltung (*digital divide*). Und diese Spaltung betrifft nicht nur Entwicklungs- und Schwellenländer. Seit den beiden UN-Weltgipfeln zur Informationsgesellschaft (WSIS – *World Summit on the Information Society*) in den Jahren 2003 in Genf und 2005 in Turin ist klar, dass es auch in Europa und in Deutschland zur gesellschaftlichen Benachteiligung durch fehlende Kompetenz bei der Nutzung digitaler Medien kommen kann. Bildungsgrad, Alter, soziales Umfeld sind nur einige der Faktoren, die eine solche Benachteiligung bewirken können. Sicherlich auf einem anderen Niveau – aber auch hierzulande muss darüber nachgedacht werden, wie eine solche Spaltung vermieden werden kann.

Wenn in dieser Ausgabe von LOG IN das OLPC-Projekt und sein Ansatz vorstellt werden, so kann dies nur angemessen erfolgen, wenn neben dem Konzept der Initiatoren auch die ersten Erfahrungen in beteiligten Ländern betrachtet werden. Dem dienen die Beiträge in der Rubrik „Thema“ dieses Heftes. Dort wird auch der zugrunde liegende *Konstruktionismus* problematisiert und dem allgemeineren Ansatz des *Learning by Design* gegenübergestellt. In der Rubrik „Diskussion“ wird die Frage gestellt, was vom OLPC-Konzept in die bundesdeutsche Schulwirklichkeit übertragbar wäre. Und im Teil „Praxis & Methodik“ werden die zentralen Softwarekomponenten und Anwendungen für den Unterricht vorgestellt, um Anregungen zu geben, selbst im Unterricht damit – auch mithilfe der beiliegenden LiveCD – zu experimentieren.

Joachim Wedekind
Bernhard Koerber

Das ist ja der Gipfel!

didacta und CeBIT 2009
in Hannover

Unter dem Motto „Der Bildungsgipfel im Flachland“ wurde die *didacta 2009* in Hannover beworben. Rund 74000 Besucher kamen vom 10. bis zum 14. Februar auf das Messegelände und informierten sich bei 718 Ausstellern auf einer Ausstellungsfläche von rund 30000 Quadratmetern über aktuelle Produkte, Trends und neue bildungspolitische Ansätze. Bereits die Eröffnung war hochkarätig besetzt: Der Ministerpräsident von Niedersachsen, Christian Wulff, und die Bundesministerin für Bildung und Forschung, Annette Schavan, betonten den Stellenwert der Bildung in der bundespolitischen Landschaft. „Lehren heißt nicht, ein Glas zu füllen, sondern ein Feuer zu entzünden“, so Ministerpräsident Wulff in seiner Eröffnungsrede, und er machte den Lehrerinnen und Lehrern Mut: „Sie machen den besten Beruf, weil wir Ihnen das wertvollste anvertrauen – unsere Kinder.“ Die Bundesministerin proklamierte das Ziel, bis zum Jahre 2015 zehn Prozent des Bruttosozialeinkommens in die Bildung zu investieren. Auf der anderen Seite mahnte der Präsident des didacta-Verbandes, Wassilios E. Fthenakis, an, dass „wir noch weit entfernt sind von einem Bildungssystem des 21. Jahrhunderts. Es gleicht einem Haus mit 16 Stockwerken, in dem jedes Stockwerk von einem anderen Architekten gebaut worden ist.“

Mit diesen Worten gerüstet konnte sich der Besucher durch die vier von der *didacta* belegten Hallen bewegen und feststellen, dass die Bedeutung des Schulbuchs ungebrochen ist. So dominierten die Schulbuchverlage auch wieder die Ausstellungshallen mit ihren zahlreichen Angeboten. Dabei bleibt die Zukunft des Schulbuchs ungewiss, denn in zunehmendem Maße beklagen die Verlage die länderspezifischen Besonderheiten in den Bildungsangeboten. Die dadurch verbundene Verteuerung der Schulbuchproduktion ist auf Dauer nicht zu vertreten.

Allerdings wächst der Markt für digitale Medien und Lernsoftware beträchtlich. So wurde das „Ende der Kreidezeit“ propagiert und als Alternative das interaktive Whiteboard vorgestellt. Gleich eine Reihe von Ausstellern brachte diese „Tafel von morgen“ als multimediales Arbeitsinstrument mit, nicht zuletzt deshalb, weil gerade Niedersachsen – nach Aussage des Ministerpräsidenten – flächendeckend dieses High-Tech-Instrument in den Schulen des Landes einführen will. Was dann allerdings damit angestellt werden soll, darüber schweigen sich die Anbieter, aber auch Bildungspolitiker und Didaktiker aus. Hier gilt es nun, sinnvolle Konzepte zu entwickeln, um nicht einem wie immer gearteten „neuen“ Frontalunterricht Vorschub zu leisten (siehe auch den Beitrag „Digitale Fenster öffnen“ von Thomas Iser, S. 73 ff. in diesem Heft).

Nintendo kam zur *didacta 2009*, um den Einsatz von Spiel-Konsolen wie *Wii* und *Nintendo DS* für den Schulunterricht zur Diskussion zu stellen. So wurde z. B. Lernsoftware zum Sprachenlernen als „mobiles Sprachlabor“ vorgestellt oder „Mathe mit Motivation und Spaß“ proklamiert. Unter der Frage „Wii fit sind unsere Schüler?“ wurde ein etwas anderer Sportunterricht vorgestellt, natürlich mit der Wii-Konsole.

Die Lernsoftware für Whiteboards und Nintendo liefern Schulbuchverlage. Damit verstärkt sich ein Trend des Zusammenwachsens der Medien, wobei das klassische Lehrbuch mit digitalen Anwendungen und internetbasierten Angeboten verbunden wird. „Ziel dieser Modernisierung ist es, die Arbeit der Pädagogen zu entlasten und den Lerner zu unterstützen“, erläuterte Andreas Baer vom *VdS Bildungsmedien* die neue Entwicklung in der Branche.

Ein Messegipfel war sicherlich auch die Verleihung des *digita 2009*, eines Preises für digitale Bildungsmedien. In seinem Gastvortrag anlässlich der Preisverleihung stellte der Hauptgeschäftsführer des BITKOM, Bernhard Rohleder, drängende Fragen: „Womit setzen sich Schüler heute auseinander? Was dürfen sie im Internet tauschen? Was sind erlaubte und was sind verbotene Downloads? Was kann man in Communitys bei schülerVZ, bei studiVZ preisgeben? Wo beginnt die Eigenverantwortung beim Schutz der Privatsphäre? Wo beginnt die Verantwortung auch für andere? Das sind ganz neue Lerninhalte in der Schule. Wir brauchen neue und wir brauchen auch junge Lehrer in den Schulen. Wir plädieren dafür, dass gerade in der aktuellen Situation Stellen geschaffen werden für junge Lehrer, die selbst sogenannte digital natives sind, die mit den digitalen Technologien aufgewachsen sind, die auf Augenhöhe darüber mit den Schülern sprechen können und insofern glaubwürdige Vermittler eines neuen Wissens sind.“

Nur drei Wochen nach der *didacta* war das Messegelände in Hannover plötzlich wieder gefüllt, zwar nicht so komplett, wie man es die Jahre zuvor gewohnt war, aber die Leitmesse der Informationselektronik *CeBIT* rief, und die Interessierten kamen, auch nicht mehr so zahlreich, wie man es in den Jahren zuvor gewohnt war. Zwanzig Prozent weniger Besucher stellten jedoch den Erfolg der Messe nach den Analysen der Veranstalter nicht infrage. *Cloud Computing*, *Webciety*, und *Mobilität* waren die zentralen Begriffe und Themen der Messe.

Cloud Computing kann vereinfacht wie folgt beschrieben werden: Die IT-Infrastruktur, speziell die Software, wird durch den Anwender nicht mehr selbst betrieben, sondern über einen oder mehrere Anbieter gezielt bezogen. Die Anwendungen und Daten befinden sich nicht mehr auf dem lokalen Rechner, sondern – bildlich gesprochen – in der Wolke (engl.: *cloud*). Dies lässt sich leichter veranschaulichen, wenn man sich vor Augen hält, dass in gängigen Abbildungen das Internet immer als „Wolke“ visualisiert wird. Der Zugriff auf diese entfernten Systeme erfolgt über ein Netzwerk, und so werden Cloud-Computing-Dienste in der Regel über das Internet bezogen.

Bei dem Schlagwort *Webciety* – ein CeBIT-Kunstwort, zusammengesetzt aus den Begriffen *Web* und *Society* – geht es um alles rund um Anwendungen und Inhalte des Web

2.0. Das Kunstwort soll den Einfluss des Internets (insbesondere des WWW) auf inzwischen alle Bereiche des gesellschaftlichen und wirtschaftlichen Lebens erfassen. Ob Kontaktpflege, E-Learning oder Arbeitsabläufe im Geschäftsumfeld – überall spielt das Internet inzwischen eine zentrale Rolle. Soziale Netzwerke, Blogs oder Wikis, Microblogging, Webvideos und Photosharing – all die zugehörigen Aktivitäten spielen sich im virtuellen Raum ab, ohne physisch greifbar zu sein. Auf diese veränderten Anforderungen reagierten die CeBIT-Veranstalter in diesem Jahr erstmals mit der „Webciety Area" in Halle 6, einem neuartigen Messekonzept. Alles, was die Aussteller mitbringen mussten, war ein Laptop. „Lästige Möbel, doofe Prospekte, alles was man als normale Internetfirma sowieso nicht hat, braucht man da auch nicht", beschrieb Web-2.0-Aktivist Sascha Lobo sein Webciety-Konzept.

Anknüpfend an *Webciety* wird die *Mobilität* zum zentralen Anliegen der Internetgemeinde. Haben wir früher bei Messeberichten von der CeBIT die zunehmende Miniaturisierung in den Mittelpunkt gestellt, dürfte diese Phase nahezu abgeschlossen sein. Zwar steigt die Leistung der Prozessoren immer noch nach dem Moore'schen Gesetz, doch die Gerätemaße bleiben zunehmend konstant und sind durchgängig „pockettauglich". An jedem Ort zu jeder Zeit alles erreichen zu können, ist das Prinzip der Stunde. Wenn man sich allein den Erfolgsweg des *iPhones* ansieht, dann weist das in die Richtung der „Web-Native-Generation". Doch mobiles Internet ist ja nicht Neues: Mit WAP (***W**ireless **A**pplication **P**rotocol*) fing es an bei Geräten mit geringer Bildschirmauflösung und langsamer Datenübertragung (HSCSD und GPRS). Alle Mobilfunkanbieter sind inzwischen zu preiswerten Daten-Tarifen als Flatrate übergegangen, und Netbooks, die auf der CeBIT 2008 ihre Geburtsstunde in Deutschland erlebten, haben sich in dem einen Jahr in beträchtlicher Anzahl verbreitet. Und es gibt kaum mehr Mobilfunktelefone (Handys) ohne Internetzugang, und nicht zuletzt RIM (***R**esearch **i**n **M**otion*) zeigte sich auf der CeBIT mit *Blackberry*-Produkten als Platzhirsch der Branche.

Gesellschaft im Wandel – das ist nun wirklich der Gipfel von Hannover 2009. An dieser Stelle muss noch einmal der Hauptgeschäftsführer des BITKOM – dem fachlichen Veranstalter der CeBIT –, Bernhard Rohleder, mit seinem Gastvortrag zur Verleihung des *digita* auf der *didacta* zitiert werden: „Wir müssen […] dafür sorgen, dass neue Medien in die Schulen kommen, und wir müssen ebenso dafür sorgen, dass neue Lerninhalte in die Schulen kommen. Wir müssen auch dafür sorgen, dass wir mehr und besser ausgebildete Lehrer haben, die sich mit den neuen Medien und den neuen Lerninhalten auskennen. Ich bin der festen Überzeugung, dass wir unser Bildungssystem umbauen müssen zu einem Bildungsdienstleistungssystem, das Schüler und Studenten ihr Leben lang begleitet und ihnen ihr Leben lang Lehr- und Lernangebote macht, gerade auch vor dem Hintergrund, dass das, was wir heute lernen – insbesondere in den technischen Bereichen – in drei, in fünf Jahren völlig veraltet ist."

Pe

Programmierunterricht in welcher Programmiersprache?

Eine Streitfrage

Das neue Ergänzungsfach *Informatik* darf seit dem Schuljahr 2008/2009 erstmals an Schweizer Gymnasien angeboten werden. Im Mittelpunkt steht dabei der Programmierunterricht, der die Fähigkeit zum Problemlösen fördern soll. In welchen Programmiersprachen sollen die Begriffe vermittelt werden? Die Meinungen der Fachleute gehen auseinander. Die folgenden Gedanken sind als Entscheidungshilfe gedacht. Einführungen ins Programmieren bewähren sich übrigens bereits an der Grundschule.

Auf die Frage nach der am besten geeigneten Lehrsprache gibt es nicht eine einzige richtige Antwort. Je nach Zielsetzung und Zielgruppe kommen mehrere Programmiersprachen in Frage. Eine solche Sprache sollte möglichst einfach erlernbar und klar aufgebaut sein.

Programmierunterricht möglichst unabhängig von einer Programmiersprache

Laut Jürg Nievergelt vom Departement Informatik der ETH Zürich heißt *Programmieren im Unterricht* (nicht jedoch in der beruflichen Praxis), „einen Algorithmus in einer beliebig vorgegebenen formalen Notation auszudrücken". Diese künstliche Sprache muss natürlich genügend mächtig sein. „Der Schwerpunkt liegt also auf der Reduktion und der Formalisierung eines intuitiv verstandenen Konzepts zu einem formal definierten Objekt. In diesem Sinn ist Programmieren teilweise unabhängig von der Sprache, in der das Programm am Schluss geschrieben wird und abläuft. Um die Unabhängigkeit von der Sprache verständlich zu machen, ist es lehrreich, denselben (einfachen) Algorithmus in verschiedenen Notationen auszudrücken. Damit das Programmieren und nicht die gewählte Sprache im Vordergrund steht, soll die Sprache so einfach wie möglich sein."

Es ist laut Nievergelt lehrreich, zusätzlich zu einer bekannten Programmiersprache einfache künstliche Sprachen zu erfinden und zu verwenden, die auf einen besonderen Anwendungsbereich – z. B. einfache Grafik, Mustervergleich (z. B. in Zeichenketten), Textformatierung – zugeschnitten sind. Man finde solche Mini-Sprachen als Untermengen von bekannten Programmiersprachen, oder man könne sie auch selbst erfinden. Carl August Zehnder und Hans Hinterberger von der ETH Zürich teilen Nievergelts Meinung.

Grundsätzlich soll der Programmierunterricht soweit möglich nicht von einer vorgegebenen Programmiersprache abhängen. Wichtig sind die grundlegenden Konzepte, die in allen verbreiteten Programmiersprachen vorkommen. In der

Ausbildung kommt man jedoch kaum um die Wahl einer Programmiersprache herum. Denn Programmieren ist schließlich eine praktische Tätigkeit, bei der eine konkrete Problemlösung in ein funktionsfähiges Programm umgesetzt und erprobt werden soll, und zwar in einer bestimmten Programmiersprache.

LOGO für den Einstieg, PASCAL und DELPHI für Anfänger und Fortgeschrittene

Nach Jürg Gutknecht, Vorsteher des Departements Informatik der ETH Zürich, „ist die Frage nach der Programmiersprache auch bei betont konzeptioneller Ausrichtung des Anfangsunterrichts in Programmierung wesentlich mitentscheidend für dessen Erfolg oder Misserfolg. JAVA ist bestimmt besser geeignet als C oder C++, aber man erkauft sich damit zwei gewichtige Nachteile: erstens die kryptische Notation und zweitens die Betonung auf ‚Programmierung im Großen', d. h. auf der globalen Programmorganisation in Form von Klassen und Objekten anstatt auf der Programmierung von Algorithmen und Datenstrukturen. Es ist, als würde man die Lehre in Elektrotechnik mit den Maxwell-Gleichungen beginnen. Übrigens ist das Szenarium der Neuen Mathematik der Siebzigerjahre des letzten Jahrhunderts mit Mengenlehre als zentralem Thema der mathematischen Ausbildung durchaus vergleichbar mit dem Bestreben, die ganze Programmierung auf den Begriffen *Klasse*, *Objekt* und *Methode* aufzubauen. Als geeignete Alternative zu JAVA aus der PASCAL-Familie wäre DELPHI zu nennen. Noch heute programmieren übrigens russische Weltmeister in Wettbewerben der amerikanischen ***A**ssociation for **C**omputing **M**achinery* (ACM) mit dieser didaktisch aufgebauten Programmiersprache."

Juraj Hromkovic, Inhaber der Professur für Informationstechnologie und Ausbildung der ETH Zürich, schließt sich dieser Ansicht an. „Aus fachdidaktischen Gründen lautet unsere Empfehlung PASCAL und DELPHI." Die ETH führt seit Jahren öffentliche Lehrveranstaltungen („Programmieren für alle") für Jugendliche und Erwachsene durch. Dabei haben sich LOGO (für den Einstieg) und PASCAL (für Anfänger und Fortgeschrittene) bewährt.

Kinder finden Gefallen am spielerischen Programmierunterricht

An der ETH Zürich gibt es regelmäßige Kurse in LOGO für Kinder. Nach Hromkovic geht es dabei vor allem darum, „das Verständnis für die grundlegenden Programmierkonzepte zu vermitteln, insbesondere für den systematischen, modularen Entwurf von Programmen". Die Schildkröten- oder Igelgrafik von LOGO und SUPERLOGO rege dabei die Kinder an.

Die Kinder können das Tier (Schildkröte oder Igel) durch Befehlseingaben so bewegen, dass Zeichnungen entstehen. Dabei lassen sich die Auswirkungen ihrer Eingaben verfolgen und Fehler berichtigen. Auf diese Weise kommen die Kinder schnell zu einem Erfolgserlebnis. SUPERLOGO enthält alle maßgebenden Bestandteile von Programmiersprachen wie Bedingungen, Schleifen, Prozeduren, Variablen. Mit diesen Hilfsmitteln werden vielfältige geometrische Figuren wie Treppen, Vielecke, Kreise oder Blumen erstellt. Dabei kommen auch Wiederholungen (Iterationen) vor.

Die Lehrmethode Hromkovics ist überdies auf die Förderung von Mädchen ausgerichtet, mit erfreulichem Erfolg. Wie die Erfahrungen mit den LOGO-Kursen zeigen, könnte man mit der Informatikausbildung gleich früh beginnen wie mit dem Rechenunterricht. Programmieren erhöht die Fähigkeit, sich klar und unmissverständlich auszudrücken, und unterstützt die Entwicklung des logischen Denkens. In Ländern Osteuropas beginnt der Programmierunterricht übrigens bereits in der Grundschule.

Umfrage an verschiedenen Universitäten

Laut Walter Gander von der ETH Zürich sollte die Programmiersprache plattformunabhängig sein und auf allen Computern verwendet werden können (Windows, Mac-OS, Linux). Zudem sollte sie preiswert, am besten aus dem öffentlichen Bereich (*public domain*) sein. Neue, billige Minilaptops (wie z. B. Asus EeePC) würden bald die jetzt noch gebräuchlichen Grafiktaschenrechner, die auch algebraisch rechnen, ersetzen. Gander stellt zudem die Frage, ob eine Entwicklungsumgebung wie *NetBeans* nötig sei oder ob es nicht genüge, Skripte wie in *Matlab* oder *SciLab* zu verfassen.

Peter Müller von der ETH Zürich möchte den Studierenden ein Werkzeug in die Hand geben, mit dem sie eindrucksvolle Programme (mit grafischen Bedienoberflächen usw.) schreiben können. „PASCAL scheint mir da etwas antiquiert. Auch mit modernen objektorientierten Sprachen lassen sich die zentralen Konzepte schrittweise einführen. Man kann z. B. den Schülerinnen und Schülern ein Skelett eines JAVA-, C#- oder EIFFEL-Programms vorgeben, in dem sie zuerst einmal nur kleinere Abschnitte imperativisch programmieren. Danach kann man Prozeduren einführen, indem man mehrere Methoden in einer Klasse implementiert. Dieser Ansatz ermöglicht es, später noch weiterführende Konzepte zu vermitteln und spannendere Programme zu schreiben. Ich denke, Schüler lassen sich mit modernen Techniken leichter motivieren, und Informatik soll schließlich als modern und spannend wahrgenommen werden!"

Die ETH Lausanne hat eine Reihe von Lerneinheiten und Werkzeugen für den Unterricht von JAVASCRIPT entwickelt (http://lti.epfl.ch/). Nach Claude Petitpierre sind Konzepte zwar wichtiger als Programmiersprachen. Das dürfe aber nicht dazu verleiten, jedes Konzept mit der bestgeeigneten Sprache einzuführen. Denn so würden die Studierenden viel Zeit aufwenden, um verschiedene Sprachen zu lernen. Die Nutzung einer einzigen Sprache erlaube die Beschränkung auf die Konzepte und ihre Einbettung in einen größeren Zusammenhang. Laut Petitpierre lassen sich Konzepte zudem nur dann verstehen, wenn es gelingt, die Algorithmen für den Rechner verständlich zu

machen. Die gewählte Sprache müsse daher vollständig, ausführbar und ablauffähig sein. JAVASCRIPT eigne sich gut für Lehrzwecke, müsse aber in eine HTML-Umgebung eingebunden werden, damit ansprechende Ergebnisse erzielt würden.

Helmar Burkhart vom Departement Informatik der Universität Basel sieht zwar die Notwendigkeit von Sprachempfehlungen, möchte aber in erster Linie die Konzepte der Informatik insgesamt stärker herausgearbeitet wissen. „Programmieren ist wichtig, aber formale Beschreibungen sind auch in anderem Zusammenhang, etwa bei Datenbeschreibungen mittels XML, unabdingbar. Auch sind beim Programmieren algorithmische Belange wichtiger als syntaktische Details.“ Im Weiterbildungsprogramm für das Ergänzungsfach *Informatik* gibt er den Lehrkräften unterschiedliche Paradigmen mit in den Schulalltag. Die Beschreibung von künstlichen Pflanzen mittels formaler Grammatiken, das spielerische, ereignisorientierte Programmieren mit SCRATCH, das intuitive Kennenlernen von objektorientierten Techniken mit JAVA/BLUEJ bis hin zum skriptorientierten Paradigma in PYTHON sieht er insgesamt als interessante Folge an, die wiederkehrende Konzepte aufzeigt. „Wurde der Computer lange als eigenständiges Instrument mit lokaler Datenhaltung gesehen, sind heute eine Einbindung ins Internet und die globale Datenhaltung im Web eine Selbstverständlichkeit. Sprachen, die hierzu nicht Schnittstellen bieten, werden über kurz oder lang als altmodisch gelten“, findet Burkhart.

Das Institut für Informatik der Universität Zürich setzt nach Harald Gall zusätzlich zu LOGO, PASCAL und DELPHI auf SMALLTALK (SQUEAK) und VISUALBASIC. Für Abraham Bernstein vom gleichen Institut sollte es eine Sprache sein, die einfach, aber trotzdem sauber strukturiert ist und mit der man ziemlich schnell ansehnliche Ergebnisse erhält. Es gehe jedoch nicht nur um das Programmieren, sondern um die Denkweise von Informatikfachleuten.

Geht es vor allem um Algorithmen und Datenstrukturen, so schlägt Oscar Nierstrasz vom Institut für Informatik und angewandte Mathematik der Universität Bern eine herkömmliche prozedurale Sprache (wie PASCAL) vor. Wenn Kenntnisse zur Modellierung vermittelt werden sollen, zieht er jedoch eine objektorientierte Sprache wie SMALLTALK, PYTHON oder RUBY vor. PYTHON habe eine einfachere Syntax als RUBY. Die Universität Bern habe bei Kindern und Jugendlichen gute Erfahrungen mit der grafischen Skriptsprache SCRATCH gemacht. Die Ergebnisse seien sofort sichtbar, was die Motivation fördere.

Matthias Hauswirth von der Universität der italienischen Schweiz in Lugano möchte noch folgende visuelle Sprachen einbeziehen, die eigens für die Ausbildung entwickelt wurden: ALICE, GREENFOOT (BLUEJ) und AGENTSHEETS.

Schlussfolgerungen

Die Auffassungen über die Ausgestaltung des Programmierunterrichts an Gymnasien sind geteilt. Allgemein wird zu Recht Wert auf die Vermittlung von *Grundlagen* des Programmierens gelegt, denn die Entwicklung der Programmiersprachen steht nicht still. Ein Teil der Fachleute bevorzugt möglichst verständliche, klar aufgebaute Programmiersprachen. Für sie steht das Programmieren von Algorithmen und Datenstrukturen im Vordergrund. Der andere Teil befürwortet neuere, objektorientierte Sprachen, die sich besonders für die Modellierung eignen.

Im Mittelpunkt des Programmierunterrichts sollte die Problemlösung und nicht die Programmiersprache stehen. Für die Ausbildung gibt es mehrere geeignete Schulsprachen. Kindgerecht sind etwa LOGO, SCRATCH und SQUEAK. Für Anfänger und Fortgeschrittene werden PASCAL, DELPHI oder VISUALBASIC empfohlen. Anspruchsvoll ist JAVA, eine objektorientierte, sehr umfangreiche Programmierumgebung. Für die Lernenden ist es hilfreich, wenn sie zu Hause für die Übungen die gleiche Umgebung wie in der Schule verwenden können. Es ist daher zweckmäßig, wenn die Programmiersprache betriebssystemunabhängig, frei zugänglich und kostenfrei ist.

Die Wahl der Programmiersprache sollte nach Carl August Zehnder den Informatik-Lehrkräften überlassen werden. Walter Gander würde es begrüßen, wenn man sich

http://www.abz.inf.ethz.ch/

ABZ – das Ausbildungs- und Beratungszentrum für Informatikunterricht der ETH Zürich.

http://www.educ.ethz.ch/

EducETH – das Bildungsportal der ETH Zürich.

auf eine Sprache einigen und diese empfehlen würde. Die Vorteile wären beim Informationsaustausch evident, und zudem wäre auch die Erstellung von Lehrmitteln einfacher und attraktiver. Nach Hans Hinterberger hängt die Wahl der Sprache sowohl vom Profil der Schule als auch von den zu vermittelnden Konzepten (Objektorientierung ja oder nein?) und der verfügbaren Unterrichtszeit ab. Es sei – wie die Erfahrungen mit PASCAL und DELPHI deutlich machen – sehr vorteilhaft, wenn Auszubildende vom Neuling bis zum professionellen Programmierer die gleiche Sprache und die gleichen Konzepte anwenden. Durch den Transfer von Wissen und Konzepten werde die Programmierkompetenz auf diese Weise stark gefördert.

E.Tutorial: Rechnergestützte Einführung in die Programmierung

Hans Hinterberger vom Institut für Computational Science der ETH Zürich bietet rechnergestützte Einführungen ins Programmieren mit TURBO-PASCAL/DELPHI sowie mit JAVA an, sogenannte *E.Tutorials* (http://www.cta.ethz.ch/etutorial_katalog).

ABZ: Ausbildungs- und Beratungszentrum für Informatikunterricht (ETH Zürich)

Um den Ausbildungsnotstand in der Schweiz zu lindern und den Nachwuchs zu fördern, hat der Lehrstuhl für Informationstechnologie und Ausbildung an der ETH Zürich ein Ausbildungs- und Beratungszentrum für Informatikunterricht (ABZ) gegründet. Es bietet u.a. Informatikunterricht an ausgewählten Gymnasien und Veranstaltungen für die Weiterbildung von Informatik-Lehrkräften an. Das ABZ hat einen ausführlichen Lehrplan für das Ergänzungsfach *Informatik* erarbeitet, und es führt Programmierkurse für Kinder und Jugendliche durch. Die ETH bietet zudem für das neue Ergänzungsfach Informatik einen besonderen Zertifikatslehrgang „Informatik für Lehrkräfte“ an. Weitere Angaben sind auf der Webseite http://www.abz.inf.ethz.ch/ zu finden. Das ABZ ist Teil des Kompetenzzentrums „Lernen und Lehren in Technik, Naturwissenschaften und Mathematik“ der ETH Zürich.

Herbert Bruderer
E-Mail:
herbert.bruderer@inf.ethz.ch

Literatur

Hromkovic, J.: Lehrbuch Informatik – Vorkurs Programmieren, Geschichte und Begriffsbildung, Automatenentwurf. Wiesbaden: Vieweg + Teubner, 2008.

Hromkovic, J.: Sieben Wunder der Informatik – Eine Reise an die Grenze des Machbaren. Wiesbaden: Vieweg + Teubner, 2009.

Hromkovic, J.: Algorithmic Adventures – From Knowledge to Magic. Heidelberg u.a.: Springer, 2009.

Webseiten zur Informatikausbildung an der ETH Zürich (Auswahl)

Ausbildungs- und Beratungszentrum für Informatikunterricht:
http://www.abz.inf.ethz.ch/

Ausstellung von (technischen) Maturaarbeiten:
http://www.ethtools.ethz.ch/projects/AMA

Bildungsportal der ETH Zürich:
http://www.educ.ethz.ch/

Einsatz von Informatikmitteln:
http://www.evim.ethz.ch/

Elektronisches Tutorial:
http://www.cta.ethz.ch/

ETH unterwegs (Wanderausstellung für Kantonsschulen):
http://www.ethtools.ethz.ch/projects/enr

Frauenförderung – Schnupperstudium Informatik:
http://www.frauen.inf.ethz.ch/

Informatikstudium:
http://www.maturandeninfo.ethz.ch/

Informationstage für Maturandinnen und Maturanden:
http://www.maturandeninfo.ethz.ch/news/infotage_lang

Jahr der Informatik – informatica 08:
http://www.informatica08.inf.ethz.ch/

Kompaktkurse für Informatiker:
http://www.inf.ethz.ch/kk

Open Class:
http://www.abz.inf.ethz.ch/

Professur für Informationstechnologie und Ausbildung:
http://www.ite.ethz.ch/

Rechnergestützter Programmierunterricht:
http://www.cta.ethz.ch/

Schweizer Informatikolympiade:
http://www.soi.ch/

Studienwochen für Mittelschülerinnen und Mittelschüler:
http://www.ethtools.ethz.ch/projects/stdw

Zertifikatslehrgang Informatik für Lehrkräfte:
http://www.inf.ethz.ch/zertifikat

Alle Internetquellen wurden zuletzt am 20. März 2009 geprüft.

Sprechende Katze und Zeichenschildkröte

Erste Schritte im visuellen Programmieren mit *Turtle Art* und SCRATCH

von Rüdeger Baumann

Der Computer ist der Proteus unter den Maschinen,
sein Wesen ist die Universalität.
Weil er sich in tausend Gestalten verwandeln
und tausend Rollen annehmen kann, vermag er es,
tausend verschiedenen Ansprüchen zu genügen.

Seymour Papert

Der XO-Laptop bietet so großartige Möglichkeiten zur kreativen Arbeit mit dem Computer, dass man diese nicht nur Kindern in Äthiopien, Brasilien oder Nepal gönnen, sondern auch hierzulande die Gelegenheit ergreifen möchte, Kinder mit der „grünen Maschine" (oder – auf einem herkömmlichen PC – mit der als Ersatz angebotenen LiveCD) bekannt zu machen. Dies dürfte der Intention des OLPC-Projekts wohl nicht entgegenstehen, da ja inzwischen ein nicht unbeträchtlicher Teil der dritten Welt „in diesem unserem Lande" ansässig geworden ist.

Eine besonders attraktive Möglichkeit, den Computer als Werkzeug zum entdeckenden und intuitiven Problemlösen einzusetzen, bietet die – auch in dieser Zeitschrift schon häufig beschriebene und empfohlene – Zeichenschildkröte (*turtle*) als Teil der Programmiersprache LOGO. Auf dem XO-Laptop steht sie nun in einer visuellen Programmierumgebung zur Verfügung und erlaubt auf besonders intuitive Weise die Erstellung von Grafiken auf dem Bildschirm (siehe Wedekind/Kohls, 2009, in diesem Heft, S. 45 ff.). Ganz nebenbei fällt damit ein Stück propädeutischer Algorithmik mit ab. Ähnliches gilt für das Entwicklungssystem SCRATCH, das übrigens nicht nur auf dem XO-1 zur Verfügung steht.

Was sagen die Bildungsstandards?

Während die derzeitigen Schulbücher für die Sekundarstufe I die Algorithmik (im Zusammenhang mit dem objektorientierten Programmieren) erst für Klasse 10 vorsehen, erwähnen die *Bildungsstandards Informatik* elementare Algorithmen schon wesentlich früher (AKBSI, 2008, S. 31):

„Schülerinnen und Schüler der Jahrgangsstufen 5 bis 7

- ▷ benutzen die algorithmischen Grundbausteine zur Darstellung von Handlungsvorschriften,
- ▷ entwerfen Handlungsvorschriften als Text oder mit formalen Darstellungsformen,
- ▷ entwerfen und testen einfache Algorithmen".

Die Handlungsvorschriften sind im vorliegenden Fall Anweisungen zur Steuerung der Zeichenschildkröte oder eines anderen, eventuell selbstentworfenen, Objekts (etwa in SCRATCH oder *Etoys*).

„An verschiedenen Beispielen erleben die Schülerinnen und Schüler, wie ein Herausfinden dieser algorithmischen Grundbausteine hilft, Handlungsvorschriften auch in Programmen realisieren zu können. Eine Kodierung in einer Programmiersprache ist in dieser Altersstufe nicht als Mindestforderung aufzustellen. Wichtiger ist es, Standardformulierungen („wenn …, dann …") immer wieder zu nutzen oder durch grafische Darstellungen ein Verständnis für die algorithmischen Grundbausteine zu erreichen. Einfache Werkzeuge oder spielerische Programmierumgebungen können dieses Vorgehen eher unterstützen" (AKBSI, 2008, S. 31 f.).

In der visuellen Programmierumgebung von *Turtle Art* oder SCRATCH sind diese Grundbausteine als beschriftete Kacheln (mit Leerstellen für Zahlen und andere Parameter) sichtbar und können mit der Maus auf die Arbeitsfläche gezogen und aufeinandergestapelt werden. Natürlich soll es hier nicht darum gehen, den Algorithmusbegriff zu definieren oder die Eigenschaften von Algorithmen aufzuzählen – das ist Schulbuchweisheit. Vielmehr sollen – als Propädeutik des Algorithmierens – Handlungsvorschriften ersonnen und mithilfe der vorgegebenen algorithmischen Bausteine realisiert werden.

Im Folgenden wird über erste Schritte berichtet, die zwei Jungen und ein Mädchen (der Jahrgangsstufen 4, 6 und 7 – künftig als „Schüler" bezeichnet) aus einer Familie „mit Migrationshintergrund" in einer Woche der Osterferien mit der LiveCD (*Turtle Art* und SCRATCH)

gemacht haben. An je zwei Vormittagsstunden wurde mit der Zeichenschildkröte gearbeitet; nach vier Tagen stand SCRATCH auf dem „Stundenplan“. Die Schüler, auch der jüngste, erwiesen sich im Umgang mit Computerspielen aller Sorten und auch mit dem Internet als äußerst versiert. Die meisten Spiele auf der LiveCD kannten sie schon; die einfacheren taten sie als „baby“(haft) ab.

Arbeit mit der Zeichenschildkröte

Nach einem kurzen Bericht über OPLC und XO-1, illustriert mit Bildern aus dem Internet, kam die Frage, was man alles „damit machen“ könne. Auf die Stichwörter „Spielen – Lernen – Programmieren“ meinte der Jüngste (Chalat), dass er schon immer habe programmieren lernen wollen. Beim Startbildschirm mit den im Kreis angeordneten Angeboten wünschte das Mädchen (Dilsah, 6. Klasse Gymnasium) zu wissen, was sich hinter der Katze (Symbol für SCRATCH) verberge. Ich vertröstete sie auf später und lenkte die Aufmerksamkeit zunächst auf das Schildkrötensymbol. Die Frage: „Wieso Schildkröte?“ wurde mit dem Hinweis beantwortet, dass diese für irgendein kleines Tier (etwa einen Igel oder dergleichen) stehe, das sich auf dem Bildschirm bewegen könne und so steuern lasse, dass eine sichtbare Spur verbleibt. Bezüglich der Beispiele hielten wir uns an das klassische Buch von Abelson/diSessa (1981).

Nach den ersten Versuchen zur Fortbewegung der Schildkröte in senkrechter und waagerechter Richtung fragte Dilsah, ob diese sich auch in schräger Richtung bewegen könne, sie möchte nämlich ein A („für Freundin Annika“) und ein D (für den eigenen Namen) zeichnen lassen.

Beispiel 1 (Buchstabenpaar): Die Schildkröte soll ein A und ein daran angehängtes D zeichnen.

Das „Dach“ des A ließ sich sehr einfach erstellen; der Winkel für den waagerechten Querstrich bereitete jedoch einige Schwierigkeiten (hier wurde zunächst etwas planlos herumprobiert); seine Länge ergab sich dann aber nach wenigen Versuchen. Das D als Halbkreis war leicht (Bild 1).

Übungsaufgaben

1. Zeichne nacheinander die Buchstaben deines Namens und beschreibe dein Vorgehen im Journal. (Anleitung: Die Kachel mit der Aufschrift „pu“, von *pen up*, dient zum Anheben des Malstifts; die Aufschrift „pd“, von *pen down*, bedeutet dessen Absenken.)
2. Die Schildkröte soll eine Jing-Jang-Figur zeichnen (Anleitung: siehe Google-Bilderdienst. Für einen Kreis, der im Gegenuhrzeigersinn, also links herum, durchlaufen wird, musst du ein Minuszeichen vor die Winkelangabe setzen.)

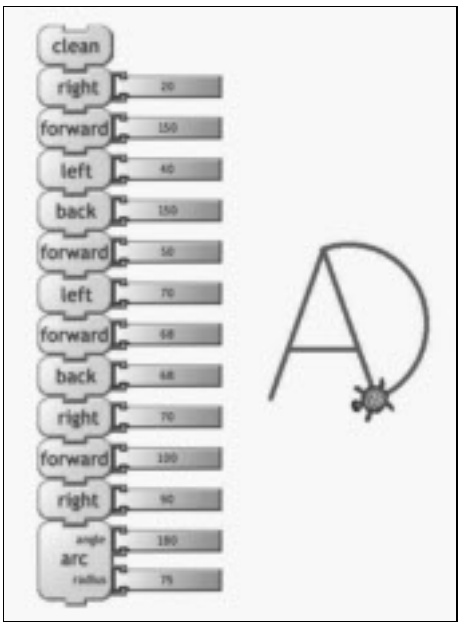


Bild 1: „Geradeaus-Programm“ (Beispiel 1).

Wiederholungsanweisungen und Unterprogramme

Als Nächstes boten sich Wiederholungsanweisungen an. Auf diese wurden die Schüler ganz von allein dadurch geführt, dass zur Ausführung einer Anweisungsfolge auf eine der Kacheln geklickt werden muss, was zu wiederholtem Klicken einlädt.

Beispiel 2 (Kreisfigur, siehe Abelson/diSessa, 1981, S. 12): Nach Zeichnung eines Kreises soll die Schildkröte sich um einen bestimmten Winkel nach links wenden und diese beiden Tätigkeiten solange wiederholen, bis die Figur „fertig“ ist.

Zunächst wurde die Abweichung auf 40 Grad eingestellt, nach neunmaligem Klicken war die Figur fertig. „Warum gerade neunmal? – Stellt mal einen anderen Winkel ein!“ Bei 50 Grad beispielsweise erwiesen sich 24 Kreise als nötig. Nun sollten die Schüler die Anzahl der Wiederholungen (Kreise) bei gegebenem Winkel vorhersagen. Zu diesem Zweck wurde die Kachel für die Zählschleife eingeführt: damit brauchte die vermu-

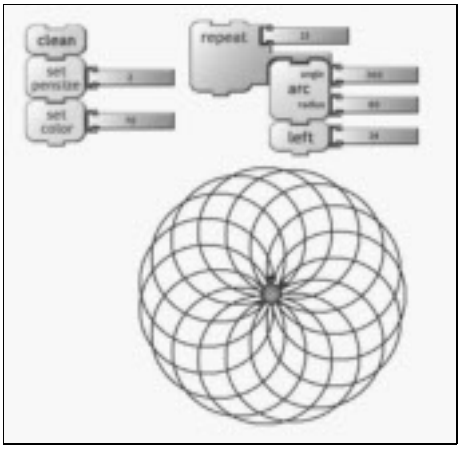


Bild 2: Wiederholung einer Anweisungsfolge (Beispiel 2).

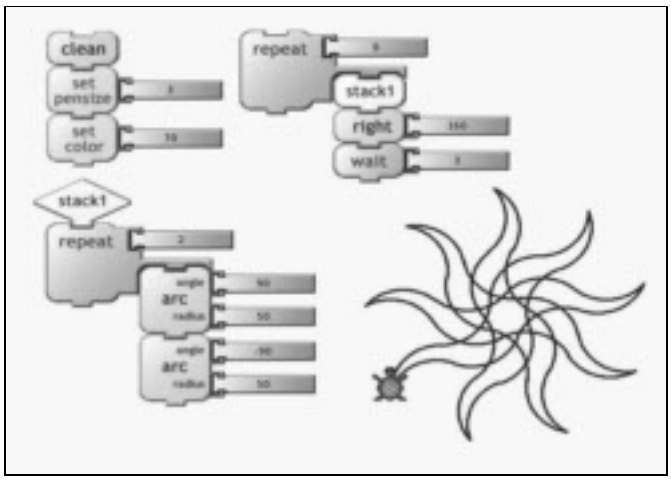


Bild 3:
Wiederholung eines Unterprogramms (Beispiel 3).

tete Anzahl nur eingetippt zu werden (Bild 2, vorige Seite).

Der Aufbau der Schleife wird übersichtlicher, wenn man die zu wiederholenden Anweisungen als „Unterprogramm“ zusammenfasst, wie folgendes Beispiel zeigt.

> **Beispiel 3** (Sonne): Eine als Sonnenstrahl gedeutete Wellenlinie soll so oft um einen bestimmten Winkel gedreht werden, bis sich die Figur schließt.

Den Anweisungsblock („Stapel“) für das Zeichnen des Sonnenstrahls kürzen wir ab, indem wir die Kachel mit der Aufschrift „stack1“ oben anfügen (Bild 3). Damit ist eine rudimentäre Form von Unterprogramm oder Prozedur (ohne Parameter) gefunden. Eine Benennung der Prozedur wie in LOGO (Abelson/diSessa, 1981, S.5) und damit die Idee: „Programmieren = Spracherweiterung“ ist hier nicht möglich. Insofern ist das visuelle Programmieren mit *Turtle Art* schwieriger, da die Prozedur bzw. das Unterprogramm sich nicht mit einem „sprechenden“ Namen aufrufen lässt. Man kann allenfalls bei der Programmbeschreibung im Journal angeben, was der Stapel (das Unterprogramm) bewirkt.

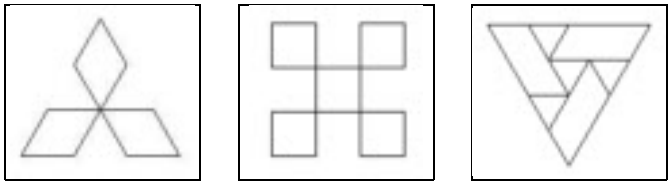


Bild 4: Figuren, die sich auf unterschiedliche Weise zeichnen lassen.

Übungsaufgaben

1. Die Schildkröte soll ein Quadrat zeichnen, sich dann um einen bestimmten Winkel drehen und diese zwei Tätigkeiten wiederholen, bis die Figur (Quadratrosette) „fertig“ ist.
2. Die Figuren von Bild 4 lassen sich auf mindestens zweierlei Arten zeichnen! (a) Drei Rauten oder drei Haken? (b) Fünf Quadrate oder vier Haken? (c) Dreiecke oder Parallelogramme?

Variablen und Wertzuweisungen (Schachtelmodell)

Um eine Figur skalierbar, das heißt in beliebiger Größe darstellbar zu machen, legt man für eine Grundgröße eine Variable fest, von der alle anderen Größen abhängen. In *Turtle Art* gibt es als Variablenersatz zwei „Schachteln“, auf die man schreibend und lesend zugreifen, die man aber nicht benennen kann.

> **Beispiel 4** (Schachteldreiecke): In ein gleichseitiges Dreieck soll ein Dreieck gleicher Art, aber halber Größe eingefügt werden – und so weiter.

Die Schildkröte zeichnet durch Wiederhole dreimal {vor(s), links(120)} ein gleichseitiges Dreieck, wobei für die Variable s eine „Schachtel“ (box1 oder box2) verwendet wird, die zu Beginn einen bestimmten Zahlenwert, etwa 100 erhält. Nun wird die Wertzuweisung s ← s/2 ausgeführt, indem der alte Wert in der Schachtel gelöscht und der neue hineingeschrieben wird: store in box1: box1 / 2 (Bild 5).

Wir können Variablen auch dazu verwenden, um verschiedene Versionen der gleichen Figur, z. B. eines Vielecks, zeichnen zu lassen.

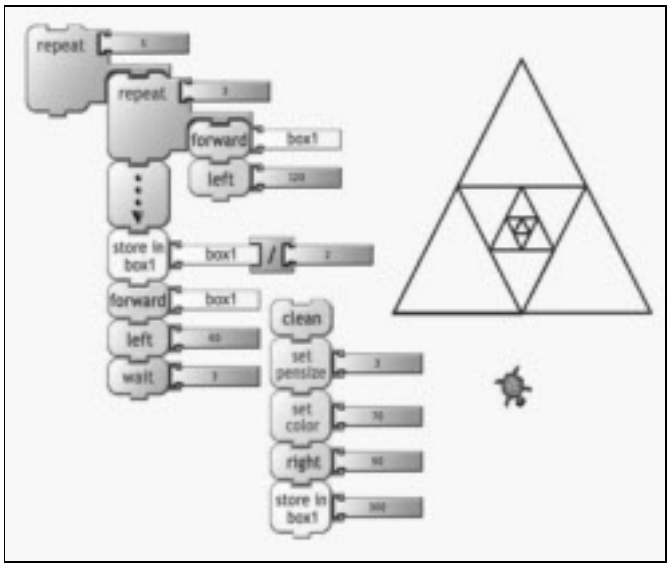


Bild 5:
Variable und Wertzuweisung (Beispiel 4).

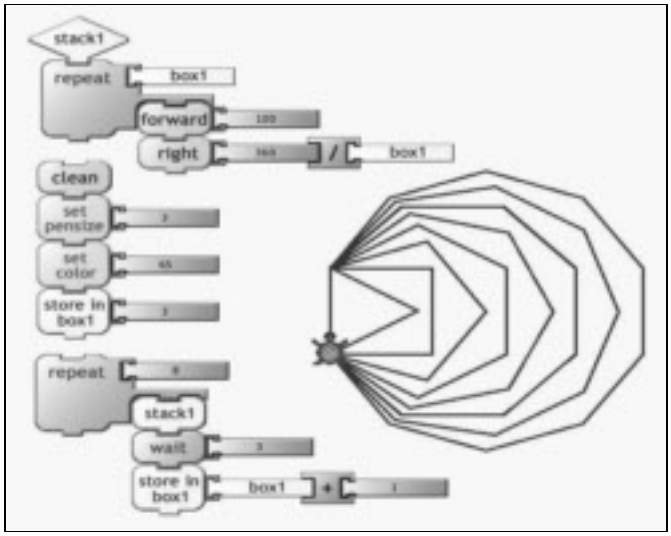
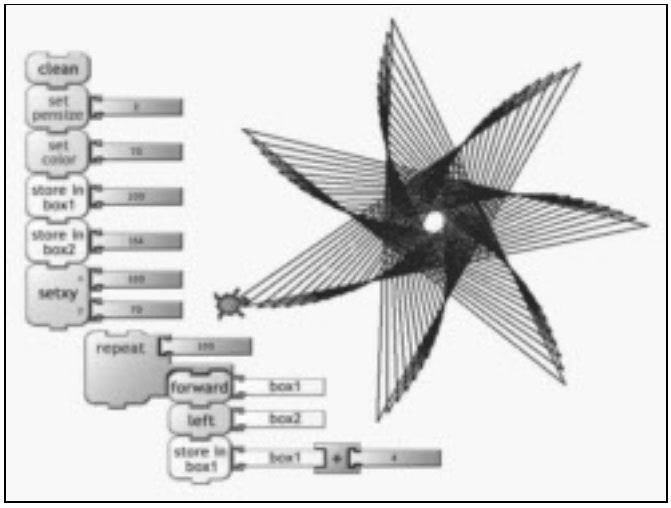


Bild 6 (oben): Unterprogramm mit laufend erhöhter Eckenzahl (Beispiel 5).

Bild 7 (unten): Variablen für Seitenlänge und Ablenkwinkel (Beispiel 6).



Beispiel 5 (Regelmäßige Vielecke): Es sollen regelmäßige Drei-, Vier-, Fünfecke usw. gezeichnet werden, die eine Seite gemeinsam haben.

Beim regelmäßigen n-Eck muss die Schildkröte sich um den Winkel 360/n Grad drehen. Wir beginnen mit n = 3 und erhöhen den Inhalt der ersten Schachtel (box1) jeweils um 1 (Bild 6).

Beispiel 6 (Polyspirale, siehe Abelson/diSessa, 1981, S. 18 f.): Die Schildkröte läuft ein Stück geradeaus, wendet sich um einen bestimmten Winkel nach links und verlängert die Vorwärtsstrecke dann um einen gewissen Zuwachs – und so weiter.

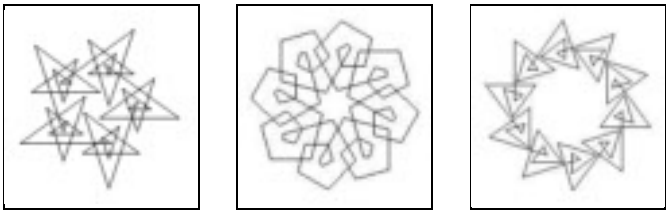


Bild 8: Spirolateralkurven. (a) n = 8, phi = 144; (b) n = 9, phi = 75; (c) n = 6, phi = 126.

Für Seitenlänge und Ablenkwinkel (box1 und box2 in Bild 7) wird je eine Variable benötigt.

Übungsaufgaben

1. Eine Spirolateralkurve (Abelson/diSessa, 1981, S. 37) entsteht dadurch, dass die Schildkröte zunächst eine gewisse Anfangsstrecke geradeaus läuft und sich dann um einen gegebenen Winkel *phi* nach links wendet. Dies wiederholt sie – jedoch mit doppelter, dann mit dreifacher usw. Streckenlänge, und zwar so oft, wie die eingegebene Anzahl n vorschreibt. Damit hat sie eine Grundfigur (Motiv) erzeugt. Dieses Motiv wird jeweils nach Drehung um *phi* solange wiederholt, bis sich die Figur schließt (s. Bild 8).
2. „Sieben Groschen“ (Bild 9, links) und Schachtel-Pentagramm (Bild 9, rechts) sollen mit der Schildkröte gezeichnet werden. (Anleitung zum Pentagramm: Die Diagonale im großen Pentagramm ist 1,62-mal so lang wie die Grundseite. Schreibe dafür vor(box1 x 162 / 100), wenn in box1 die Länge der Grundseite gespeichert ist.)

Die Katze im Computer

Gegenüber *Turtle Art* bietet SCRATCH ein wesentlich reichhaltigeres Arsenal an gestalterischen Möglichkeiten, andererseits aber auch Einschränkungen (siehe unten). Es handelt sich um eine Entwicklungsumge-

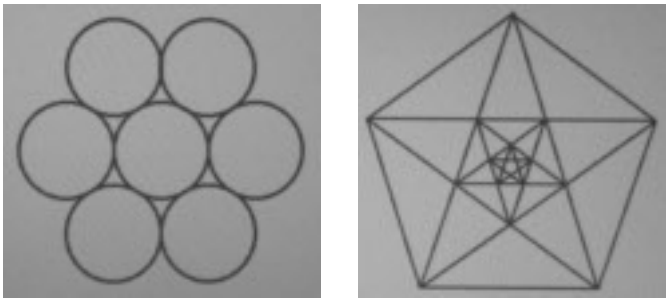


Bild 9: Sieben Groschen (links) und Schachtel-Pentagramm (rechts).

Bild 10 (oben): Der Objektbereich (Ausschnitt) von Beispiel 7.

Bild 11 (links): Die Bühne am Ende von Szene 2 (Beispiel 7).

bung, die – wegen der intuitiven Bedienung und der leichten Definition und Steuerung grafischer Objekte – zum spielerischen Experimentieren einlädt, ja geradezu verführt (vgl. Romeike, 2007; Stoll/Thalheim/Timmermann, 2008).

Fast ohne Anleitung verstanden die Schüler sofort die Einteilung der Benutzungsoberfläche in Programmierwerkzeuge (Kacheln), Programmbereich, Bühne und Objektbereich; sie konnten ohne weiteres Objekte erstellen (malen oder importieren), ihr Aussehen variieren, sie zu gegebenen Punkten gleiten lassen, Klänge beifügen usw. Mit der selbstgestellten Aufgabe (Simulation eines Boxkampfs) waren sie aber überfordert, sodass sie das Angebot, erst einmal die grundlegenden Programmiertechniken kennenzulernen, akzeptierten.

Beispiel 7 (Szenenfolge zur Erschaffung der Welt): In der (Grundschul-) Klasse, so stellten wir uns vor, sitzen Schüler und Schülerinnen christlichen, moslemischen, hinduistischen u. a. Glaubens. Um den Mitschülern Einblick in die Vorstellungswelt der eigenen Religion zu vermitteln, soll eine einfache Szenenfolge erstellt werden, die die Erschaffung der Welt schildert. In der jesidischen *Schwarzen Schrift* (siehe Issa, 2008, S. 31 f.) heißt es:

Zuerst schuf Chodeh (Gott) aus seinem gebenedeiten Schoße eine weiße Perle und er erschuf einen Vogel, dem er den Namen Anfar gab, und die Perle legte er auf dessen Rücken und sie lag vierzigtausend Jahre auf ihm. Dann erschuf er, beginnend mit dem Sonntag, täglich einen Engel und machte Tausi-Melek zu einem Großen. Hernach schuf er die sieben Himmel sowie Sonne und Mond.

Wir legen fest, dass folgende Gestalten auftreten sollen: Chodeh, Perle, Vogel Anfar, sechs Engel, Oberengel Tausi-Melek, Sonne, Mond, Himmel (als Hintergrund). Zur Darstellung der Objekte werden selbstgefertigte Fotografien verwendet, die eventuell auf dem Maltablett zu bearbeiten sind (Bild 10). Die einzelnen Szenen werden auf Papier skizziert:

- ▷ Szene 0 (Anfangsbild) mit dem Text „Die Erschaffung der Welt“ und einem jesidischen Tempel;
- ▷ Szene 1: Chodeh erschafft die Perle und spricht;
- ▷ Szene 2: Chodeh erschafft den Vogel Anfar, die Perle rollt auf seinen Rücken (Bild 11) usw.

Entscheidend ist nun die Frage, wie die Aufeinanderfolge der Szenen gesteuert wird. Wir vereinbaren, dass der Benutzer nach dem Startkommando (Anklicken der grünen Flagge am oberen Bildschirmrand) in das Geschehen nicht mehr eingreifen können soll. Die Schüler schlagen folgende Lösung vor: Die Objekte Chodeh, Perle, Vogel usw. sind zwar immer vorhanden, aber zuweilen unsichtbar („verstecke dich“), zuweilen sichtbar („zeige dich“); sie erscheinen nur nach einer ihnen eigentümlichen Wartezeit auf der Bühne (siehe Bild 12). Es gibt bei dieser sehr simplen Lösung also kein von einer zentralen Stelle ausgehendes Signal und keine Benachrichtigung der Objekte untereinander; vielmehr agiert jedes Objekt nach seinem eigenen Zeitplan. Dies erfordert vom Programmierer eine genaue Vorgabe der Wartezeiten.

Übungsaufgabe

Die Steuerung der Szenenfolge von Beispiel 7 mittels Wartezeiten ist aufwendig und umständlich. Erfin-

Bild 12: Das Programm für Chodeh (oben) und Perle (links).

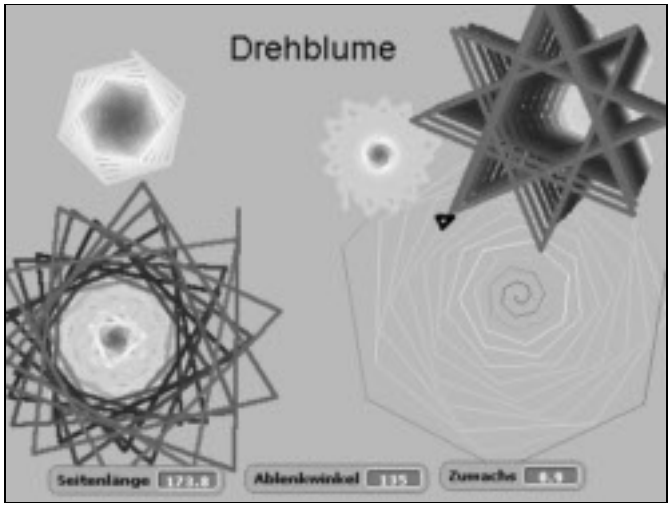


Bild 13: Zufällige „Drehblumen“ (Polyspiralen) gemäß Beispiel 6 bzw. 8.

de eine bessere Methode derart, dass die Objekte jeweils benachrichtigt werden, wenn sie sichtbar werden sollen. (Nimm gegebenenfalls das Projekt „Creation-Story“ auf der LiveCD als Vorbild – aber mache es besser!)

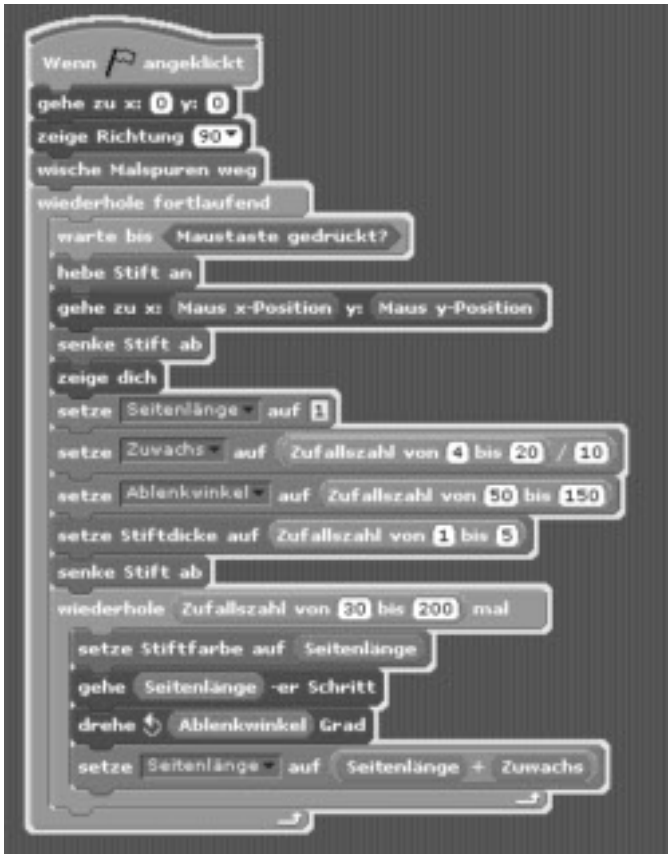


Bild 14: Das Programm zu Beispiel 8 (Drehblume).

Beispiel 8 (Drehblume): Die aus Beispiel 6 bekannten Polyspiralen sollen – bei zufällig gewähltem Zuwachs, Ablenkwinkel, Stiftfarbe usw. – an der Stelle auf dem Bildschirm erscheinen, wo der Benutzer mit der Maus hinzeigt, und zusätzlich von Klängen untermalt werden.

Wir konstruieren ein neues Objekt als kleines Dreieck auf dem Maltablett und haben damit automatisch eine Zeichenschildkröte gewonnen, da (außer Bühnenobjekten) jedes neugeschaffene Objekt von SCRATCH deren Eigenschaften und Fähigkeiten besitzt. Es braucht also nur das Programm von Bild 6 übersetzt zu werden. Die Anweisungen, d. h. die Beschriftungen der Kacheln, sind jetzt in Deutsch formuliert: Statt „forward“ heißt es „gehe …-er Schritt“ usw. Der entscheidende Unterschied zu *Turtle Art* besteht jedoch darin, dass es in SCRATCH – statt unbenannter Schachteln – Variablen gibt, deren aktuellen Wert man auf der Bühne sichtbar machen kann (Bild 13). Die Wertzuweisung wird durch Setze … auf … ausgedrückt (Bild 14). Die Schüler erlernen diesen Variablengebrauch schnell, und auch arithmetische Operationen und Zufallszahlen bereiten keine Verständnisschwierigkeiten.

Eine sehr mächtige Fähigkeit von SCRATCH besteht darin, dass man ein Objekt auf ein anderes „ausrichten“ und damit Verfolgungskurven zeichnen lassen kann.

Beispiel 9 (Ameisenrennen): Vier vom informatischen Geist inspirierte Ameisen (Annika, Bettina, Corinna und Dilsah) sitzen an den Ecken eines Quadrats. Heuschrecke Hulda gibt das Startkommando: Annika verfolgt daraufhin Bettina, diese setzt hinter Corinna her, letztere hat es auf Dilsah abgesehen, und diese auf Annika; alle rennen mit der gleichen Geschwindigkeit. Welche Spuren hinterlassen sie?

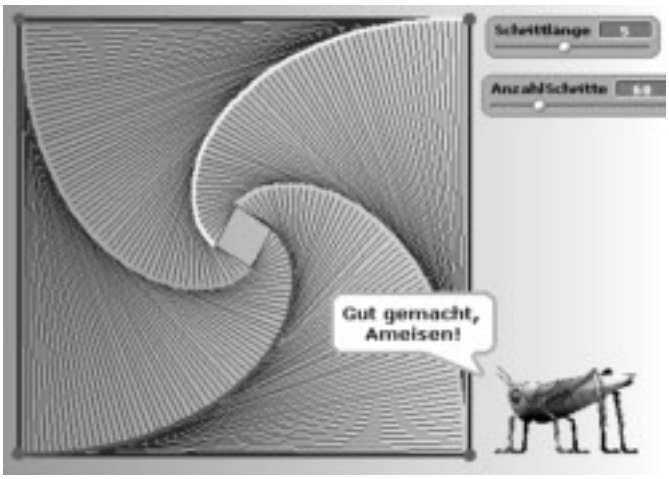


Bild 15: Nach Schluss des Ameisenrennens (Beispiel 9).

Bild 16: Programm für Ameise Annika (Beispiel 9).

Als Objekte werden zunächst vier Punkte kreiert, die auf der Bühne ein Quadrat bilden, sodann die vier Ameisen, und schließlich Heuschrecke Hulda, die die Nachricht „Start“ an alle sendet und zum Schluss einen Kommentar abgibt (Bild 15, vorige Seite). Das Programm für Annika (Bild 16) stimmt mit dem der drei übrigen Ameisen – bis auf die Änderungen bezüglich Startaufstellung und Bewegungsrichtung – überein: Prozeduren (mit Parametern), die eine solche Vervierfachung des Programmtextes entbehrlich machen würden, gibt es nicht.

Übungsaufgaben

1. Ermittle durch Experimentieren mit dem Programm zu Beispiel 9, wie viele Schritte in Abhängigkeit von der Schrittlänge bis zum Zusammentreffen nötig sind, und ersetze die Variable *AnzahlSchritte* entsprechend.
2. Lege andere Startaufstellungen fest (gleichseitiges Dreieck, regelmäßiges Sechseck, Vieleck mit besonders attraktiven Spuren).

Abschließende Bemerkungen

Die LiveCD (und natürlich erst recht der XO-1) beinhalten ein faszinierendes Angebot an Gelegenheiten zum Spielen und „Lernen durch Gestalten“ völlig unterschiedlichen Anspruchsniveaus und Motivationsvermögens. Besonders die hier näher betrachteten Möglichkeiten zur visuellen Programmierung sind sicher bereits in der Grundschule zwecks spielerischer Propädeutik des Modellierens und Algorithmierens einsetzbar. Die unterschiedliche Terminologie von *Turtle Art* und SCRATCH scheint keine Schwierigkeiten zu bereiten (da die Wirkung der Anweisungen auf dem Bildschirm unmittelbar erkennbar ist).

Viele der den XO-1 auszeichnenden Möglichkeiten zur Kooperation der Lernenden untereinander konnten in der beschriebenen Woche allerdings nicht erprobt werden, da nur ein einziges Gerät (bzw. ein PC) zur Verfügung stand. Es wurde keinerlei informatische Fachterminologie explizit eingeführt, aber informell von Variablen, Objekten, Anweisungen usw. gesprochen, sodass sich ganz nebenbei ein gewisser fester Sprachgebrauch einstellte.

Ermöglicht SCRATCH genügend „informatische Substanz“?

„Wie sollte in das Programmieren eingeführt werden?“ fragt Romeike (2007, S. 36) und plädiert für einen „kreativen Informatikunterricht“ mit SCRATCH. Solange man lediglich beansprucht, einige Animationen und Spiele zu gestalten, um den Lernenden einen „kreativen Einstieg in die Programmierung“ anzubieten, mag das (eigentlich für Grundschulkinder gedachte) Programmierwerkzeug selbst in Jahrgangsstufe 11 akzeptabel und brauchbar sein. Anders sieht es aus, wenn man als Lehrkraft verpflichtet ist, einen „Stoffverteilungsplan Informatik“ zu erstellen und sich dabei an staatliche Lehrplanvorgaben halten muss oder Bildungsstandards genügen möchte: Hier stellen sich bezüglich der Eignung von SCRATCH erhebliche Zweifel ein.

„Sieht man sich im World Wide Web um, so findet man viele SCRATCH-Projekte, die eine sicher nett anzusehende Animation liefern, aber teilweise wenig ‚informatische Substanz‘ enthalten“, stellen Stoll/Thalheim/Timmermann (2008, S. 82) zu Recht fest. Ihr Unterrichtsversuch beansprucht dagegen, „informatisch substanziell“ zu sein, indem er sich akribisch an die Bildungsstandards Informatik zu halten vorgibt; doch wird er diesem Anspruch nicht in jeder Hinsicht gerecht. Denn er vernachlässigt u. a. eine äußerst wichtige Kompetenz: die Fähigkeit zum Strukturieren und damit zur Reduktion von Komplexität. „Die Schülerinnen und Schüler aller Jahrgangsstufen strukturieren Sachverhalte durch zweckdienliches Zerlegen und Anordnen“ (AKBSI, 2008, S. 50). Was heißt das in der Algorithmik? Abelson/Sussman (1998, S. 57) geben eine Antwort:

> Wir können von einer leistungsfähigen Programmiersprache erwarten, dass sie es ermöglicht, Abstraktionen durch Zuweisung von Namen an häufig verwendete Strukturen zu bilden, und dann mit den Abstraktionen direkt weiterzuarbeiten. Mit Prozeduren haben wir diese Möglichkeit. Deshalb gibt es in allen außer den ganz einfachen Programmiersprachen Mechanismen zur Definition von Prozeduren.

In *Turtle Art* ist eine rudimentäre prozedurale Abstraktion möglich (siehe Beispiel 3), in SCRATCH jedoch nicht einmal dies. So lassen sich z. B. in SCRATCH die Figuren von Bild 4, Bild 8 und Bild 9 nur mit mehrfach geschachtelten Wiederholungsanweisungen erzeugen: unnötige Komplexität, die durch prozedurale Abstraktion vermieden würde. Das Gleiche gilt für die

unökonomische Programmtext-Vervielfachung in Beispiel 9. Die im Beitrag von Zahn (2009, in diesem Heft, S. 27 ff.) beschriebene Konstruktion von Wissen beim Lernen beruht zu einem bedeutenden Teil auf Abstraktion; diese beginnt im frühesten Kindesalter. Die Möglichkeit dazu muss in einer Programmiersprache unbedingt gegeben sein.

Zwar wird in den Bildungsstandards die prozedurale Abstraktion nicht explizit erwähnt, wohl aber an vielen Stellen die Datenabstraktion – leider ist auch diese in SCRATCH nicht möglich. Dessen Strukturierungsmittel sind Objekte, doch fehlt die Möglichkeit zur Klassenbildung, und damit die Möglichkeit zur Modellierung von lebensweltlichen Situationen (außer trivialen Animationen und Spielen), da es nur zwei Klassen (Sprite und Stage) gibt. Die Konzepte *Assoziation* und *Vererbung*, die für die objektorientierte Modellierung wesentlich sind und deren Wert begründen, fehlen. Nicht einmal das EVA-Prinzip lässt sich demonstrieren.

Fazit

Vorliegender Beitrag beschreibt zwar keinen regulären Unterrichtsversuch, lieferte aber die begründete Vermutung, dass sich *Turtle Art* und SCRATCH für einen kreativen Einstieg in die Programmierung für Kinder im Grundschulalter eignen, nicht aber für einen Informatikunterricht in der Sekundarstufe I, der den Bildungsstandards Informatik verpflichtet ist. Wenn Stoll/Thalheim/Timmermann „festhalten, dass SCRATCH eine wirkliche Alternative zu KARA und KAROL darstellt“ (2008, S. 90), so mag hier festgehalten werden, dass das Werkzeug vielleicht nicht ganz so schlecht wie diese, aber auch nicht „wirklich“ besser ist.

Nach der vorstehend geschilderten Osterwoche äußerte sich übrigens der Jüngste des kleinen Kollektivs (derzeit noch in der vierten Grundschulklasse), nun endlich „programmieren lernen“ zu wollen und auf PippyPYTHON (auf dem XO-1 bzw. der LiveCD ebenfalls vorhanden) gespannt zu sein. Das Aufeinanderstapeln von Kacheln empfand er irgendwie nicht als „richtiges Programmieren“ (woher er diese Erkenntnis hatte, blieb unbekannt). Vielleicht ein Hinweis auf PYTHON als „wirkliche Alternative“?

Rüdeger Baumann
Fuchsgarten 3
30823 Garbsen

E-Mail: baumann-garbsen@t-online.de

Literatur

Abelson, H.; diSessa, A.: Turtle Geometry. Cambridge (Mass.): MIT Press, 1981.

Abelson, H.; Sussman, G. J.: Struktur und Interpretation von Computerprogrammen. Berlin u. a.: Springer, [3]1998.

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.

Issa, Ch.: Das Jesidentum – Religion und Leben. Oldenburg: Dengeh Jesidian, 2008.

Prätorius, P.: Grafik im Anfangsunterricht. In: LOG IN, 24. Jg. (2004), Heft 127, S. 35–45.

Romeike, R.: Animationen und Spiele gestalten – Ein kreativer Einstieg in die Programmierung. In: LOG IN, 27. Jg. (2007), Heft 146/147, S. 36–44.

Stoll, Th.; Thalheim, K.; Timmermann, B.: Die Katze im Computer – Das Programmierwerkzeug SCRATCH im Informatikunterricht der Sekundarstufe I. In: LOG IN, 28. Jg. (2008), H. 154/155, S. 81–90.

Wedekind, J.; Kohls, Chr.: Programmieren mit dem XO-Laptop. In: LOG IN, 29. Jg. (2009), Heft 156, S. 45–50 (*in diesem Heft)*.

Zahn, C.: Gestaltendes Lernen – „Learning by design“ im Schulunterricht. In LOG IN, 29. Jg. (2009), Heft 156, S. 27–35 (*in diesem Heft)*.

Hardware

Der XO-Laptop im Netz

Die weitreichenden und pädagogisch wertvollen Möglichkeiten des SUGAR-Betriebssystems zum gemeinsamen Arbeiten und zur Kommunikation mit anderen XO-Laptops (siehe auch den Beitrag von Rita Freudenberg, Seite 40 ff. in diesem Heft) erfordern eine ausgefeilte und leistungsfähige Netzwerk-Architektur. Sie wird im Folgenden beschrieben.

Grundlagen der XO-Laptop-Netzwerkschnittstelle

Das integrierte WLAN-Interface mit dem Mesh-Netzwerkkonzept ist – neben dem eigens entwickelten speziellen Bildschirm – eine der großen Innovationen des XO-Laptops. Die implementierten Netzwerk-Funktionen, sowohl *Mesh-Routing* nach IEEE 802.11s als auch *Collaboration in SUGAR*, sind im Vergleich zu anderen Geräten mit ähnlichem Ansatz (z. B. CMPC, Netbook) derzeit konkurrenzlos.

Die WLAN-Schnittstelle basiert auf dem Marvell „Libertas"-Chipsatz 88W8838/88W8015 (siehe Bild 1). Sie ist ein aktives, mit einem eigenen Mikrokontroller (ARM9-basiert) versehenes Interface, mit dem sowohl Wi-Fi-Funktionalität als auch Mesh-Routing realisiert wird. Letzteres funktioniert auch unabhängig vom Hauptprozessors (AMD Geode) durch eine eigene Firmware. Das Mesh-Routing ist auch im Standby-Modus des XO-Laptops (bei geschlossenem Deckel) über Stunden voll funktionsfähig. Im Vergleich zum heute üblichen Standard ist kein klassischer Ethernet-Anschluss vorhanden. Es kann aber ein externes USB-Ethernet-Interface problemlos ergänzt werden (siehe Bild 2).

Mit dem in der verwendeten LINUX-Distribution (derzeit *Red Hat Fedora 9*) genutzten „Libertas"-Treiber werden zwei Geräte unter dem „/dev"-Baum realisiert. Neben *eth0* (klassisches Wi-Fi-Interface) existiert *msh0*, die Schnittstelle zum 802.11s-Mesh-Netz (siehe Bild 3). Wird ein unterstütztes USB-Ethernet-Interface ergänzt, erscheint zu-

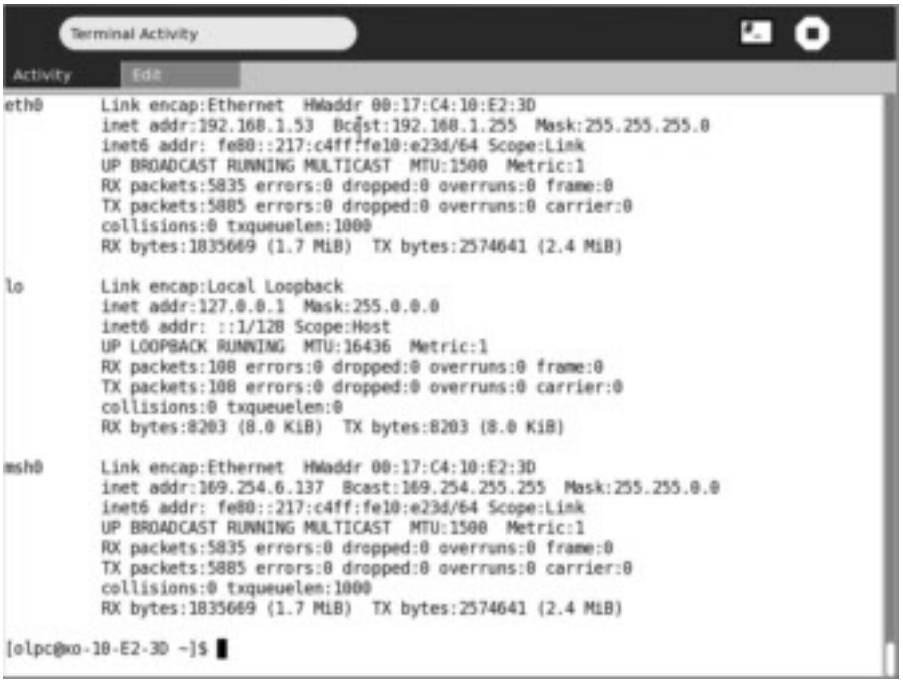
Bild 1: Marvell Wireless Chipsatz des XO.

http://bunniestudios.com/blog/images/olpc_wifidetail.jpg

Bild 2 (links): XO mit USB-Ethernet-Interface.

Foto: V. Hinz

Bild 3 (unten): Netzwerkschnittstellen des XO (ifconfig –a).



```
Terminal Activity
Activity   Edit
eth0      Link encap:Ethernet  HWaddr 00:17:C4:10:E2:3D
          inet addr:192.168.1.53  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::217:c4ff:fe10:e23d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5835 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5885 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1835669 (1.7 MiB)  TX bytes:2574641 (2.4 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:108 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:8203 (8.0 KiB)  TX bytes:8203 (8.0 KiB)

msh0      Link encap:Ethernet  HWaddr 00:17:C4:10:E2:3D
          inet addr:169.254.6.137  Bcast:169.254.255.255  Mask:255.255.0.0
          inet6 addr: fe80::217:c4ff:fe10:e23d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5835 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5885 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1835669 (1.7 MiB)  TX bytes:2574641 (2.4 MiB)

[olpc@xo-10-E2-3D ~]$
```

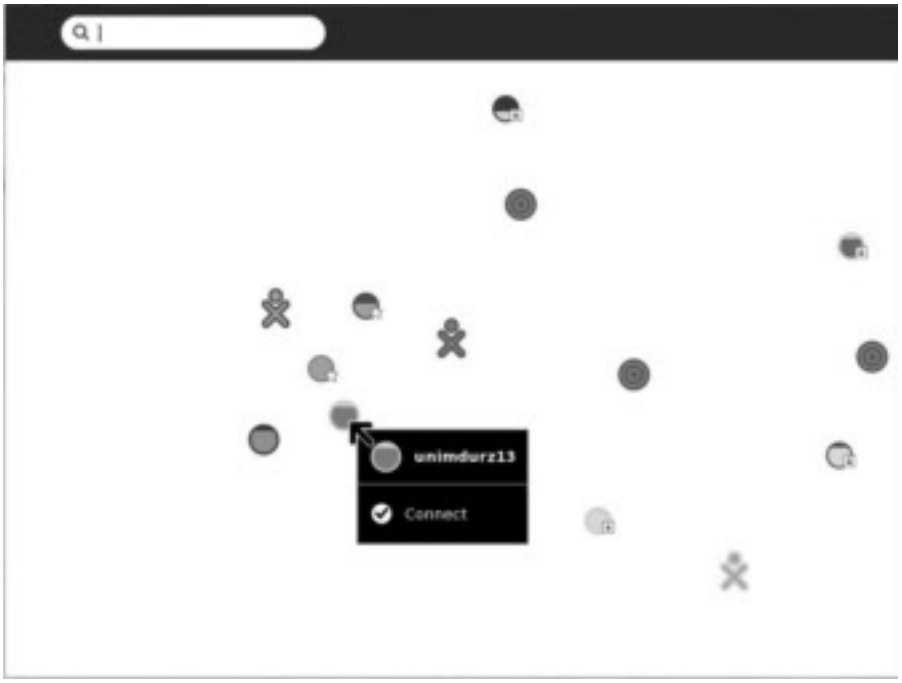


Bild 4: Nachbarschafts-Menü, Auswahl eines Wi-Fi-Zugangspunkts.

sätzlich des Gerät *eth1* (vgl. auch http://wiki.laptop.org/go/USB_ethernet_adaptors). Dieses wird bei Anschluss an ein lokales Netz automatisch per DHCP konfiguriert. Damit ist der XO-Laptop auch ohne WLAN im Netz benutzbar.

Die Geräte *eth0* und *msh0* erlauben die Nutzung der 802.11-Funkschnittstelle unabhängig voneinander. Das bedeutet, dass ein XO-Laptop sowohl über *eth0* eine Wi-Fi-Verbindung zu einem vorhandenen Zugangspunkt (*Managed*-Netz) aufbauen als auch über *msh0* an einem Mesh-Netz teilnehmen kann. Mesh-Netze sind eine erweiterte Form eines „Ad-Hoc"-Netzes. Beim XO-Laptop ist dabei in großen Teilen der IEEE 802.11s-Entwurfsstandard implementiert worden (vgl. auch Wikipedia „IEEE 802.11s"). Für das Mesh-Netz sind die überlappungsfreien WLAN-Kanäle 1 (2,412 GHz), 6 (2,437 GHz) und 11 (2,462 GHz) fest vorgesehen. Mesh- und Wi-Fi-Verbindungen sind sogar gleichzeitig möglich. Der XO-Laptop kann dann eine Internet-Verbindung in das Mesh-Netz weiterleiten (der XO-Laptop fungiert als Mesh-Portal).

Wird ein XO-Laptop in Betrieb genommen, versucht die Netzwerk-Software über das Interface *msh0* in folgender Reihenfolge eine Verbindung aufzubauen:

- ▷ Suche nach einem „School Mesh Network Portal" nacheinander auf den Kanälen 1, 6 und 11,
- ▷ Suche nach einem „XO-Laptop Mesh Network Portal" nacheinander auf den Kanälen 1, 6 und 11,
- ▷ Start eines „Simple Mesh" mittels einer IPv4LL-Adresse (169.254.x.y) auf Kanal 1.

Darüber hinaus kann über das Interface *eth0* eine Wi-Fi-Verbindung zu einem Zugangspunkt aufgebaut werden. Das geschieht durch Anwahl eines entsprechenden Symbols im Nachbarschafts-Menü (Bild 4). Das Interface *eth0* wird dabei automatisch über DHCP konfiguriert. Ist ein Wi-Fi-Netz bekannt, d.h. bereits einmal verbunden worden, erfolgt die nächste Verbindungsaufnahme automatisch. Die Symbole im Nachbarschafts-Menü bedeuten dabei Folgendes: Mesh-Netz, Wi-Fi-Netz sowie Wi-Fi-Netz (verbunden). Ein kleines Schloss an einem Wi-Fi-Symbol weist auf ein verschlüsseltes Netz (WEP, WPA), ein kleiner Stern auf ein bekanntes Netz hin.

Eine Liste bekannter Wi-Fi-Netze findet man über die Kommandozeile (Aktivität *Terminal*) in der Datei ~/.sugar/default/nm/networks.cfg. Den aktuellen Netzstatus und einige weitere interessante Informationen des XO-Laptops erhält man dort mit den Kommandos ifconfig –a bzw. olpc-netstatus –n (vgl. Bild 3, vorige Seite, und Bild 8, nächste Seite).

Szenarien von XO-Laptops im Netz

Mehrere XOs in Reichweite – ohne Zugangspunkt

Werden mehrere XOs innerhalb ihrer Netz-Reichweite, die durch die speziellen Antennen-Ohren verhältnismäßig groß ist, eingeschaltet, so erfolgt der Aufbau eines IPv4LL-Netzes (***IP version 4 local link***; siehe Bild 5). Das Interface *msh0* wird automatisch mit einer IPv4LL-Adresse konfiguriert. Alle XOs befinden sich damit in einer Broadcast-Domain (IP-Adressen 169.254.x.y/16). In der SUGAR-Benutzungsoberfläche sind nach einer kurzen Zeit die jeweils anderen XOs im Nachbarschafts-Menü als X sichtbar (siehe Bild 4). Die SUGAR-Funktionen „Geteilte Aktivitäten" (siehe unten) sind möglich,

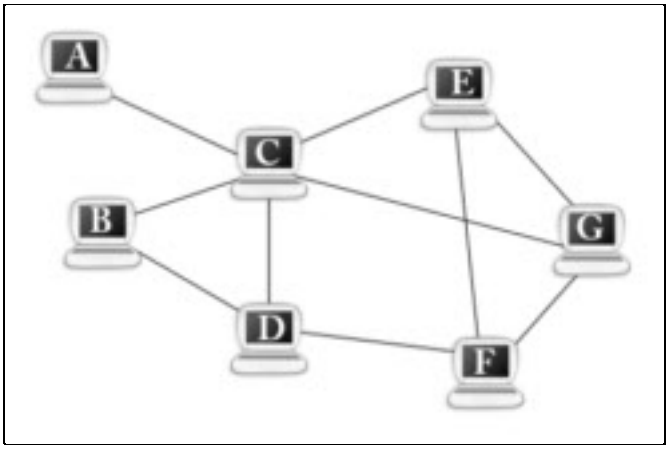


Bild 5: Simple Mesh.

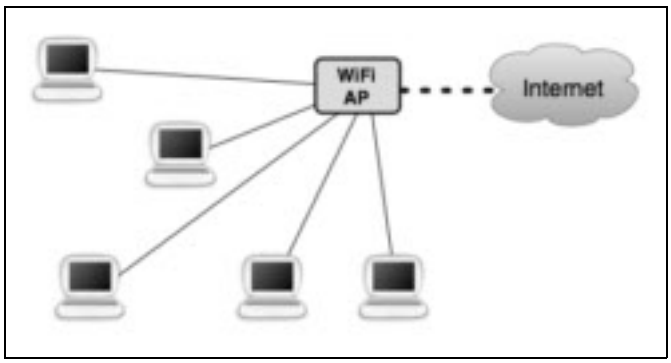


Bild 6: XOs im Wi-Fi-Netz.

ein Internetzugang jedoch nicht (Telepathy Connection Manager Salut; vgl. http://wiki.laptop.org/go/Telepathy_Salut).

Die speziellen Funktionen des Mesh-Netzes ermöglichen ein für die höheren Schichten transparentes Routing auf MAC-Ebene – zum Beispiel (siehe Bild 5, vorige Seite) könnte also der XO-Laptop A von dem XO-Laptop F über die dazwischen liegenden Computer erreicht werden. Hier sind etwa die Pfade A–C–D–F, A–C–G–F oder A–C–E–G–F möglich. Das Mesh-Routing innerhalb der Firmware bewertet dabei mögliche Pfade selbstständig und wählt für die Netzpakete jeweils den kürzesten aus. Fällt ein Pfad aus, wird eine Alternative benutzt. Diese dazwischen liegenden XOs können sich dazu im Standby-Zustand befinden, da das aktive WLAN-Interface das Routing weiter ausführt (siehe oben). Die Kommunikation von XOs über größere Entfernungen als die WLAN-Reichweite zwischen zwei XOs ist ebenfalls möglich. Befinden sich n Kommunikationspartner in linearer Reihenfolge, ist theoretisch eine (n–1)-fache Reichweite möglich.

Verfügt ein XO-Laptop über einen Internetzugang (Wi-Fi-Zugangspunkt, Modem etc.), kann man diesen zu einem Mesh-Portal (MPP) machen. Damit ist ein Internetzugang für die anderen, im gleichen Mesh-Netz befindlichen XOs ebenfalls möglich. Diese Funktion wird derzeit nicht offiziell unterstützt und ist nur über Eingaben auf der Kommandozeile zugänglich. Dazu erweisen sich tiefergehende LINUX-Kenntnisse als vorteilhaft.

Bild 7 (rechts): XOs am Mesh-Portal (Schulserver).

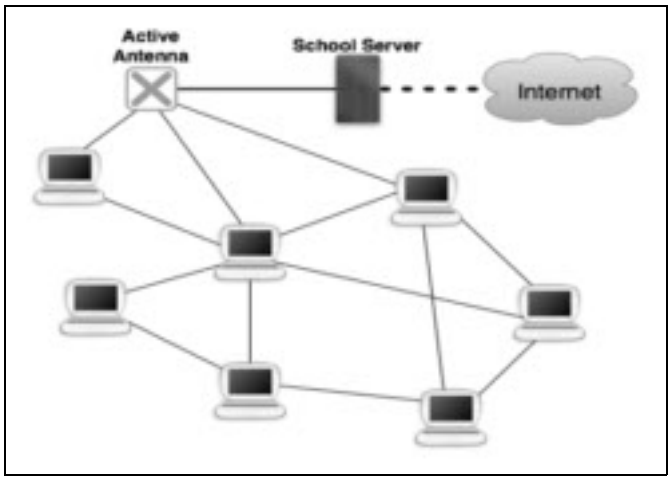


Bild 8 (unten): Netzwerkstatus des XO-Laptops am Mesh-Portal (olpc-netstatus –n).

Ein oder mehrere XOs in einem Wi-Fi-Netz

Ein XO-Laptop kann über einen Wi-Fi-Zugangspunkt Zugang zu einem lokalen Netz bzw. zum Internet bekommen (siehe Bild 6). WEP und WPA werden dabei unterstützt. Der Laptop wird dabei vom Zugangspunkt mit einer IP-Adresse konfiguriert. Liegen die vergebenen Adressen bezüglich der verwendeten Netzmaske in einer Broadcast-Domain, sind sie untereinander im Nachbarschafts-Menü sichtbar (Telepathy CM *Salut*). Das ist sinnvollerweise an öffentlichen Hot-Spots oft nicht gegeben. Die Sichtbarkeit über Router bzw. über das Internet hinweg ist unter Verwendung eines Jabber-Servers realisierbar (Telepathy CM *Gabble*). Dann ist die gemeinsame Teilnahme an einer Aktivität auch über das Internet hinweg möglich.

Mehrere XOs an einem Mesh-Portal

Das Szenario im Bild 7 beschreibt die Verwendung mehrerer XOs an einem Mesh-Portal (MPP). Ein Mesh-Portal kann dabei ein XS-Schulserver oder ein XO-Laptop selbst sein (siehe oben). Die

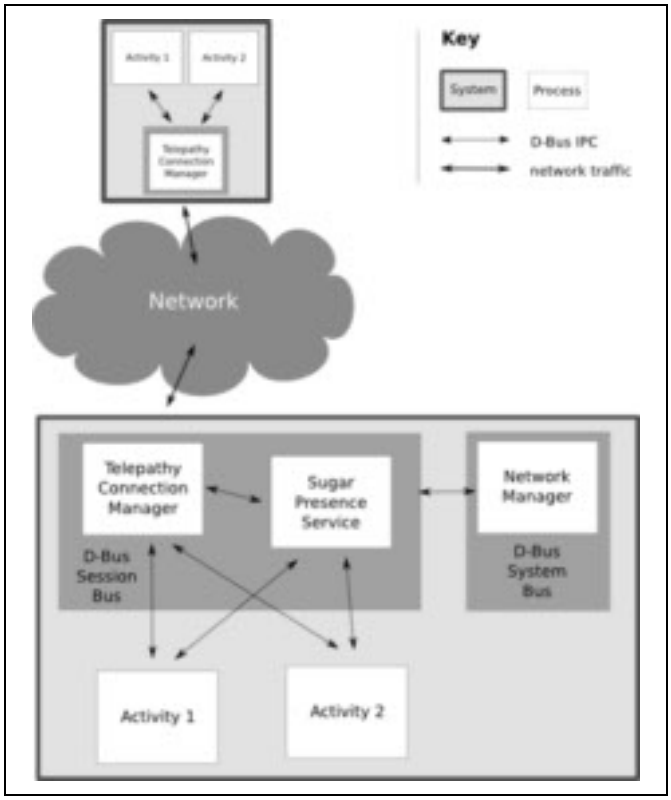


Bild 9: Struktur der Kollaborationsmöglichkeiten.

derzeit offiziell unterstützte Form ist die Verwendung eines XS-Schulservers (siehe unten). Dieser konfiguriert die teilnehmenden XOs automatisch mit Netzparametern sowie einigen zusätzlichen Informationen. Die Arbeitsmöglichkeiten entsprechen an den einzelnen XOs denen in einem einfachen Mesh-Netz, erweitert um spezielle Funktionen des Schulservers. Der Schulserver selbst implementiert einen Jabber-Server, der dann die „Aktivitäts-Teilung“ ermöglicht (ebenfalls Telepathy CM *Gabble*, vgl. Bild 8, vorige Seite).

Geteilte Aktivitäten

Eine interessante Möglichkeit der Software des XO-Laptops ist das unter dem Begriff „Kollaboration in SUGAR“ zusammengefasste gemeinsame Betreiben von Aktivitäten (*activity sharing*) von bestimmten, dafür ausgelegten Aktivitäten. Grundlagen dafür sind das *Telepathy Framework* mit den Connection-Managern *Salut* und *Gabble*, *D-Bus-Messaging* sowie der *Sugar Presence Service* (Bild 9). Für weitergehende Information zu diesem interessanten Konzept sei ausdrücklich das OLPC-Wiki empfohlen.

Die SUGAR-Oberfläche zeigt im Nachbarschafts-Menü alle im lokalen Netz verfügbaren (Telepathy CM *Salut*) bzw. über einen Jabber-Server (Telepathy CM *Gabble*) erreichbaren XOs an. Hat ein Benutzer am XO-Laptop eine Aktivität (spezielle LINUX-Applikation im SUGAR-Kontext; siehe auch den Beitrag von Rita Freudenberg, S. 40 ff. in diesem Heft) gestartet, kann er diese entweder mit der gesamten Nachbarschaft „teilen“ (*share*) oder einen Benutzer an einem anderen XO-Laptop explizit einladen (*invite to*). Nimmt dieser dann an der Aktivität teil (*join*), können beide bzw. mehrere an der Aktivität gemeinsam arbeiten. Zum Beispiel startet der Nutzer am XO-Laptop *mttcs-3* die Aktivität *Schreiben* und aktiviert in deren Aktivitäts-Menü die Option „Teilen mit meinen Nachbarn“. Die Nutzer an den anderen XOs *mttcs-2* und *mttcs-4* können dann an dieser Aktivität teilnehmen (siehe Bild 10).

Bild 10: Teilnehmen an einer Aktivität.

Bild 11: Gleichzeitiges Arbeiten in der Aktivität *Schreiben*.

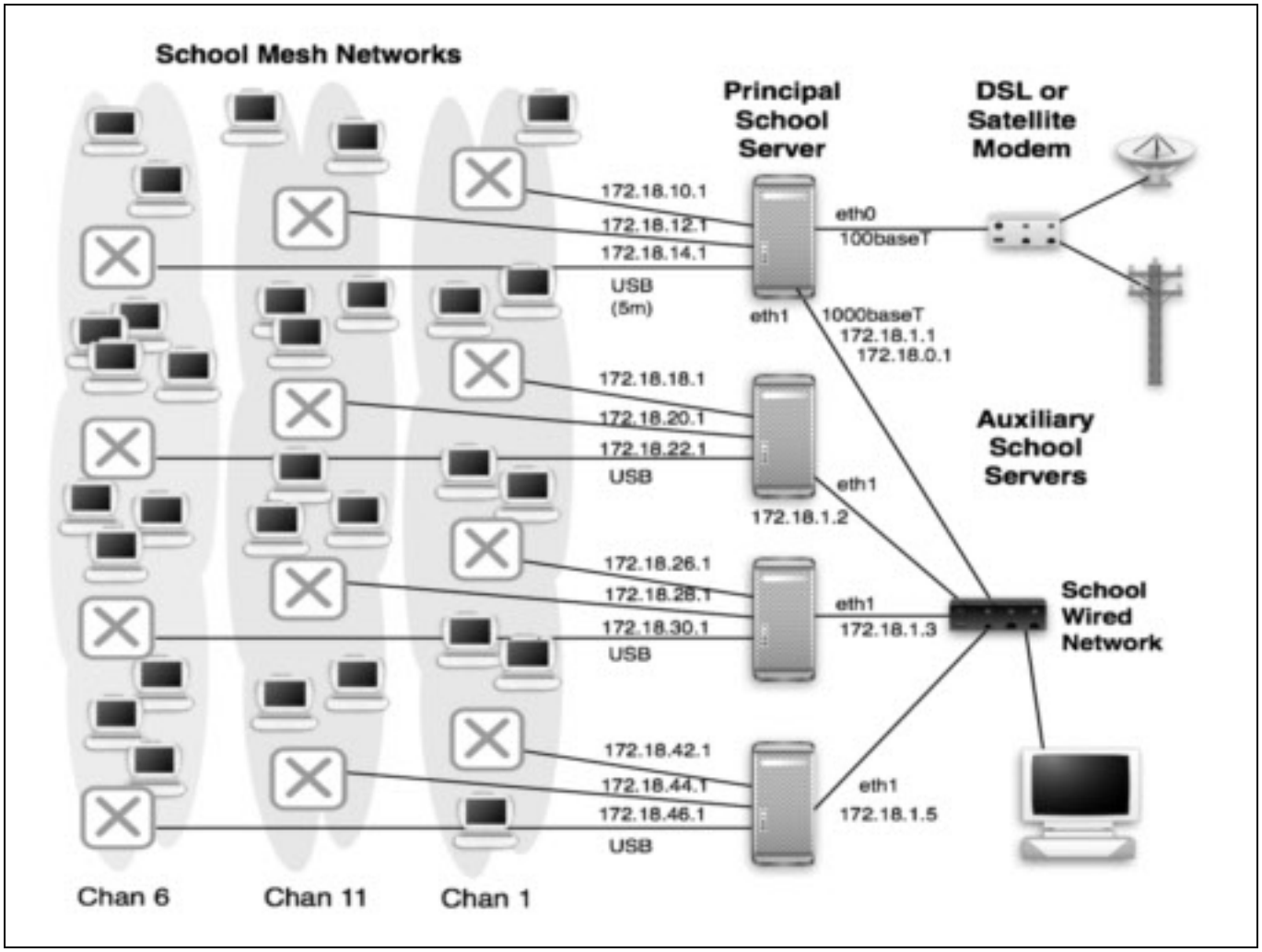


Bild 12:
Schulserver, große Ausbaustufe.

Dabei beginnt dann ebenfalls lokal eine Schreib-Aktivität, und ein gemeinsames Arbeiten am Text ist nunmehr möglich (siehe Bild 11, vorige Seite). Gewöhnliche LINUX-Applikationen sind zwar unter SUGAR lauffähig, ein Teilen von Aktivitäten geht damit jedoch nicht: dazu müssen die Applikationen angepasst, im Jargon: „sugarized" werden (siehe FLOSS Manuals).

Der XO-Schulserver XS

Der XS (***X**O school **S**erver*) ist kein Gerät an sich, sondern eine spezielle LINUX-Distribution, derzeit basierend auf *Fedora Core 9*. Die Distribution ist im Moment noch einem intensiven Entwicklungsprozess unterworfen und setzt zur Anwendung tiefer gehende LINUX-Kenntnisse sowie das Verständnis für das XO/XS-Laptop-Gesamtkonzept voraus (siehe Bild 12). Der derzeitige Entwicklungsstand der Distribution ist *Version 0.5.2* (März 2009). Lauffähig ist die Distribution auf einem (fast) beliebigen x86-Standard-PC. Der Autor verwendet zum Test einen alten PII-266-MHz-Laptop mit 228 MByte RAM (Bild 13, nächste Seite).

Zur Realisierung des Mesh-Portals ist ein Marvell *Libertas* 88W8838 basiertes WLAN-Interface zwingend notwendig. Dieses wird als Zubehör zum Schulserver als *Active Antenna* von OLPC angeboten (in Bild 13, Mitte oben). Die *Active Antenna* (Marvell Modul analog XO-Laptop und integrierte Antenne) ist in einem zum Design des XO-Laptops passenden Gehäuse untergebracht. Ein 5 m langes USB-Kabel dient als Anschlusskabel an den Schulserver und versorgt die Antenne mit Strom. Die derzeitige XS-Version 0.5.2 unterstützt bis drei Antennen für die WLAN-Kanäle 1, 6 und 11. Ein umfangreicher Ausbau mit Schulservern könnte wie in Bild 13, nächste Seite aussehen.

Als Aufgaben des Schulservers XS sind die folgenden vorgesehen:

▷ *Allgemeine Netzwerkaufgaben*:
- Schulnetz-Infrastruktur (Mesh-Portal, Intranet),
- Internetzugang (über Modem, ISDN, DSL, CDMA/GPRS/3G, Satellit, …),
- HTTP-Proxy mit möglichem Content-Filter (Jugendschutz),
- Moodle-Server für Bildungsinhalte.

▷ *Dienste für XOs (nach Registrierung des XO-Laptops am XS)*:
- Dienst für geteilte Aktivitäten (*ejabberd*),
- Backup- und Restore-Dienst (Journal der einzelnen XOs),
- Installations- und Upgrade-Server,
- Aktivierungs-Server.

Foto: V. Hinz

Bild 13: Experimenteller Schulserver des Autors.

Derzeit verfügbar in der Testversion des XS sind ein NAT-basiertes Internet-Gateway über das Ethernet-Interface *eth0* (statische IP-Adresse oder automatisch mit DHCP konfiguriert). Intern stehen ein Intranet-Interface *eth1* (vergibt über DHCP Netzwerk-Adressen) sowie maximal 3 Mesh-Portale *msh0* bis *msh2* (dafür ist jeweils eine *Active Antenna* notwendig) zur Verfügung. Ein Internet-Zugang über eine 3G-Anbieter (UMTS) ist mit wenigen Eingaben auf der Kommandozeile konfigurierbar und wurde vom Autor erfolgreich getestet.

Wird ein XO-Laptop am XS über das Menü des XO-Symbols registriert, erfolgt auch eine Registrierung am integrierten Jabber-Server des XS sowie eine Aktivierung des automatischen Backup. Dabei wird das Journal jedes am XS aktiven XO-Laptops täglich einmal gesichert. Ein *Restore* ist über ein Web-Interface des Schulservers möglich, das über die Browse-Aktivität erreicht werden kann. Am Schulserver kann ein Proxy-Cache (*squid*) aktiviert werden. Damit wird die Nutzung des WWW mit vielen XOs am XS bei langsamen Internet-Anbindungen beschleunigt. Die weiteren Dienste des XS werden mit dem Fortschreiten der Entwicklung zur Version 1.0 (vgl. http://wiki.laptop.org/go/School_server) schrittweise implementiert.

Zusammenfassend kann gesagt werden, dass das XO/XS-Gesamtkonzept interessante Aspekte einer umfassenden Schul-IT-Lösung offenbart. Einen Einsatz in der deutschen Bildungslandschaft in der für den XO-Laptop empfohlenen Altersstruktur (5 bis 10 Jahre) findet der Autor empfehlenswert. Ein entsprechendes Projekt wird an der Universität Magdeburg derzeit entwickelt.

Volkmar Hinz
E-Mail: volkmar.hinz@ovgu.de

Literatur und Internetquellen

bunnie's blog – speziell OLPC XO-1:
http://www.bunniestudios.com/blog/?p=218

FLOSS Manuals für den XO und SUGAR:
http://en.flossmanuals.net/xo
http://en.flossmanuals.net/sugar

Freudenberg, R.: SUGAR – ein Betriebssystem zum Lernen. In: LOG IN, 29. Jg. (2009), Heft 156, S. 40–44 (*in diesem Heft*).

OLPC-Wiki mit diversen Informationen:
http://wiki.laptop.org/

Wikipedia – Stichwort „IEEE 802.11s":
http://de.wikipedia.org/wiki/IEEE_802.11s

Alle Internetquellen wurden zuletzt am 20. März 2009 geprüft.

Software

Die XO-LiveCD

Dieser Beitrag stellt ein OLPC-Teilprojekt – die XO-LiveCD – vor. Mit dieser CD-ROM (siehe Umschlagseite III in diesem Heft) kann die Software des XO-Laptops auch auf traditionellen Computern eingesetzt werden. Alle Software einschließlich der Lernprogramme, die im OLPC-Projekt als *Activities* bezeichnet werden, ist für jedermann offen und wird nach dem Open-Source-Modell entwickelt.

Die XO-Software

Der XO-Laptop wurde als Lern- und Unterrichtsmedium für die Anforderungen in Entwicklungsländern konzipiert. Der XO-Laptop ist ausgestattet mit Kamera, Mikrofon, Lautsprechern und einem im Licht der Sonne lesbaren Display. Mit seinen zwei Antennen verbindet sich das Gerät automatisch mit anderen Geräten in der Nähe. Dies erlaubt das gemeinsame Arbeiten an Aufgaben im Unterricht, was mit herkömmlichen Computern nur schwer erreichbar ist.

Die Benutzungsoberfläche – der SUGAR-Desktop – hat eine besonders einfache und auf die Intuition von Kindern abgestimmte Bedienphilosophie (vgl. Freudenberg, 2009, S. 40 ff. in diesem Heft). Durch seine Konzeption, Hardware-Ausstattung und Software vereint der XO-Laptop in einem Gerät:

- ▷ Lehrbücher, Lexika, Wörterbücher,
- ▷ Medien zur schriftlichen, verbalen und visuellen Kommunikation
- ▷ und eine Plattform für naturwissenschafliche Experimente.

Um OLPC-Software auch in unseren Schulen nutzen zu können, stellt sich die Frage, ob die Anschaffung des grünen XO-Laptops notwendig und sinnvoll ist. Viele Schulen verfügen bereits über eine IT-Ausstattung, und auch der Lap-

top-Markt bietet zunehmend preisgünstig mobile Geräte an (z.B. sogenannte Netbooks), die sich für eine individuelle Nutzung durch Schülerinnen und Schüler eignen. Sicherlich üben die kleinen grünen XO-Laptops auch auf unsere Kinder eine große Faszination aus. Bei einem Vergleich mit konventionellen Notebooks sollte man aber nicht vergessen, dass die Geräte für den Einsatz in Entwicklungsländern ausgelegt sind und sich nicht am Wetteifer um immer mehr Rechenleistung beteiligen.

Welche Möglichkeiten gibt es, um OLPC-Software unter der SUGAR-Bedienungsoberfläche auch auf herkömmlichen PCs zu nutzen? Man findet dazu mehrere Antworten auf den Wiki-Seiten des Projekts (siehe Internetquelle Nr. [11]). Manche Wege sind für Softwareentwickler beschrieben und erfordern LINUX-Kenntnisse. Andere Entwicklungen ermöglichen die Integration der SUGAR-Software in gängigen LINUX-Distributionen. In einer aktuellen Entwicklung arbeiten auch die *Sugar Labs* an einer eigenen SUGAR-Distribution mit dem Namen *Sugar on a Stick* [22].

Die XO-LiveCD

Eine sehr einfache Möglichkeit die Software ohne spezielle IT-Kenntnisse auszuprobieren, ist die XO-LiveCD [02]. Es handelt sich dabei um eine konfigurierbare Laufzeitumgebung für das auf dem XO-Laptop installierte Dateisystem. Im einfachsten Fall kann man den Computer direkt mit dieser CD starten. Für den Einsatz im Schulunterricht empfiehlt sich ggf. eine Installation auf der Festplatte. Anleitungen und Hilfestellungen findet man in dem auf der CD enthaltenen Dokument (XO-LiveCD.pdf) und in Beiträgen der Mailing-Liste [18].

Die XO-LiveCD startet auf allen Intel-kompatiblen PCs (siehe auch Kasten „Start der XO-LiveCD unter Windows“) einschließlich Apple Mac-Books. Mit dieser CD soll

▷ Kindern, Schülern, Lehrern und Eltern die Chance gegeben werden, an dem OLPC-Projekt teilzuhaben und die Lernsoftware auf einem normalen PC zu nutzen,

Start der XO-LiveCD

Standardmäßig ist bei neueren PCs die Boot-Reihenfolge so eingestellt, dass zuerst im CD/DVD-Laufwerk nach einem Boot-Medium gesucht wird. Legen Sie deshalb die XO-LiveCD in Ihr CD/DVD-Laufwerk ein – sollten Sie mehrere Laufwerke angeschlossen haben, dann in Ihr primäres – und starten Sie den Rechner neu.

Sollte dies nicht zum Start der LiveCD führen, so muss die Boot-Reihenfolge im BIOS geändert werden. Das BIOS (***B**asic **I**nput **O**utput **S**ystem*) ist ein Programm, das als sogenannte Firmware in einem nichtflüchtigen Speicher auf der Hauptplatine eines PC mit einem Intel-Prozessor bzw. Intel-kompatiblen Prozessor abgelegt ist und unmittelbar nach dessen Einschalten ausgeführt wird. Es ist unter anderem die Aufgabe des BIOS, zunächst die Hardware-Bestandteile des PC zu überprüfen, ihre Funktionsfähigkeit sicherzustellen und im Anschluss das Starten eines Betriebssystems einzuleiten. Zum Ändern des BIOS haben Sie zwei Möglichkeiten. Die erste Möglichkeit:

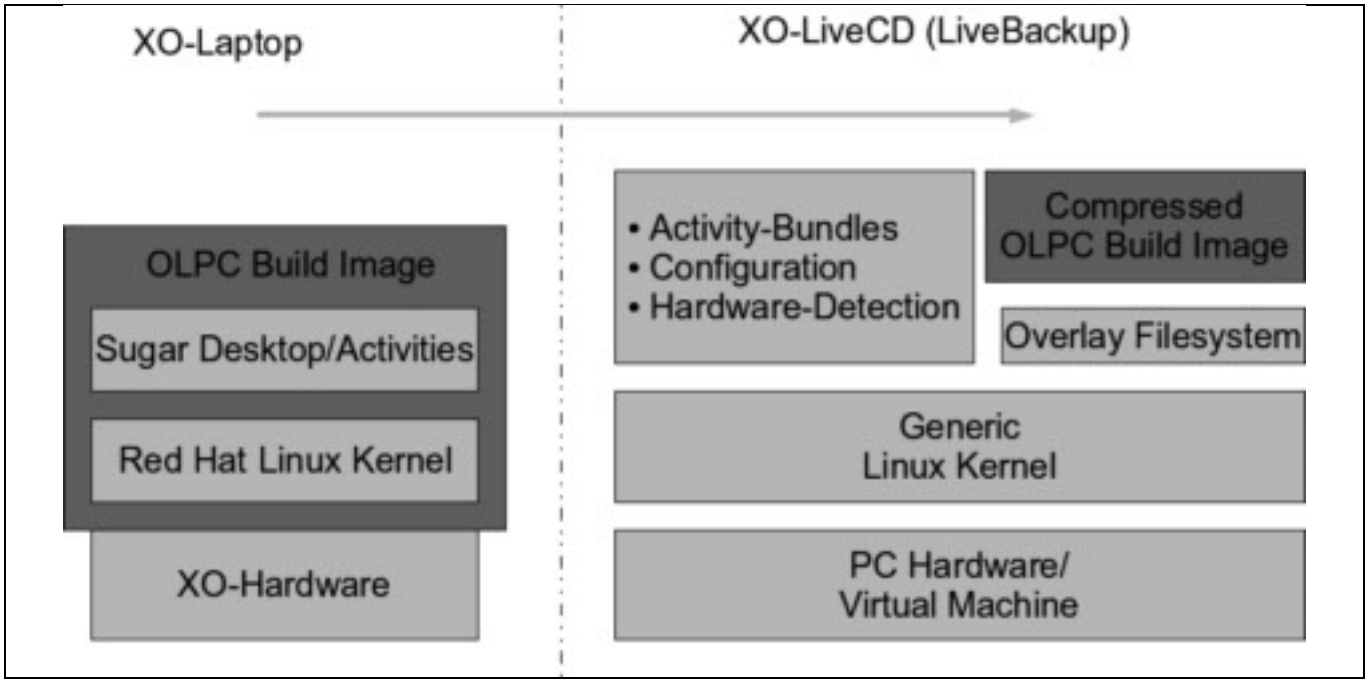
▷ Einige neuere BIOS-Versionen bieten eine eigene (Funktions-)Taste an – oft <F12> –, mit der man die Boot-Reihenfolge einmalig, das heißt nur für den nächsten Boot-Vorgang ändern kann. Wenn Sie ausnahmsweise nicht von der Festplatte, sondern eben z.B. von einer bootfähigen CD wie der LiveCD starten möchten, ist dies deshalb von Vorteil, weil Sie später nicht daran denken müssen, eine permanent geänderte Boot-Reihenfolge wieder auf das primäre Booten von der Festplatte zurückzusetzen.

Sollte auch dies nicht klappen, dann gehen Sie wie folgt vor:

▷ Starten Sie Ihren Rechner neu. Sofort nach dem Start wird Ihnen angezeigt, mit welcher Taste oder Tasten-Kombination das Menü zur BIOS-Konfiguration aufgerufen werden kann. Da die Anzeige am Bildschirm im Allgemeinen zu schnell abläuft, um sie zu lesen, können Sie sie mit der <Pause>-Taste anhalten. Im Allgemeinen wird das BIOS-Menü mit der <Entf>- bzw. <DEL>-Taste aufgerufen.
▷ Sollten Sie das BIOS mit einem Passwort geschützt haben, müssen Sie es jetzt eingeben.
▷ Anschließend erscheint ein Menü-Fenster des BIOS, z.B. mit der Überschrift „AwardBIOS Setup Utility“. Aus den darunter liegenden Menüs wählen Sie mit der <→>-Taste das Menü „Boot“ aus, was dann, wenn es durch die Pfeiltaste markiert ist, sofort angezeigt wird.
▷ Im Allgemeinen werden vier Geräte angezeigt, von denen ein Boot-Medium gestartet werden kann. Mit der <↓>-Taste markieren Sie Ihr CD/DVD-Laufwerk, in das Sie die LiveCD eingelegt haben bzw. einlegen möchten.
▷ Danach können Sie mit der <+>-Taste vom rechten Ziffernblock Ihrer Tastatur dieses Gerät in der Boot-Reihenfolge auf den 1. Platz verschieben. Entsprechend können Sie später mit der <–>-Taste das ausgewählte Gerät wieder nach unten versetzen.
▷ Anschließend wählen Sie die Option „F10 Save and Exit“ aus oder drücken die Taste <F10>. Im darauf erscheinenden Fenster werden Sie „Save configuration changes and exit?“ gefragt. Da „Yes“ voreingestellt ist, genügt es, wenn Sie die <Enter>- bzw. <Eingabe>-Taste drücken.

Der Rechner fährt weiter hoch, und die LiveCD wird gestartet, sofern sie bereits eingelegt worden ist.

Die Sitzung wird beendet, indem beim Haupt-Menü das zentrale XO-Symbol angeklickt und bei dem sich öffnenden Fenster (bitte ggf. ein wenig warten) die Option „Ausschalten“ gewählt wird.



Übertragen des XO-Laptop-Betriebssystems in eine LiveCD-Laufzeitumgebung.

- ▷ die Publizität der OLPC-Software gefördert und auch Nicht-Softwareentwicklern zugänglich gemacht werden,
- ▷ Lehrpersonen zu Modellversuchen an Grundschulen ein Anreiz gegeben werden, damit auch unsere Kinder an den technischen und pädagogischen Entwicklungen teilhaben,
- ▷ das OLPC-Projekt durch neue Inhalte und Beiträge unterstützt werden.

Die LiveCD wird ständig auf völlig unterschiedlichen Computern getestet und weiterentwickelt. Konventionelle Notebooks lassen sich mit ihr einfach in einen SUGAR-Laptop mit Dual-Boot-Option verwandeln.

Konfigurationsmöglichkeiten

Die Technik auf der XO-LiveCD bietet eine Fülle von verschiedenen Konfigurationsmöglichkeiten. Viele Optionen dienen zur Abstimmung und Optimierung der Software auf die verwendete Computer-Hardware. Aus pädagogischer Sicht interessant ist die Möglichkeit zur Auswahl von Lernprogrammen (*Activities*), die unter der SUGAR-Oberfläche den Benutzern angeboten werden. So lässt sich die XO-LiveCD auf einfache Art und Weise auf bestimmte Altersstufen abstimmen, oder die Auswahl der Software kann für bestimmte Unterrichtsziele erfolgen.

Besonders einfach gelingt eine Netzwerk-Integration (vgl. auch Hinz, 2009, S. 65 ff. in diesem Heft). Nach dem Einstecken des Ethernet-Kabels ist das System automatisch in das Netzwerk eingebunden. Befinden sich mehrere XO-Laptops bzw. XO-LiveCD-Systeme im Netzwerk, so erkennen sich die Geräte automatisch und die – in den SUGAR-Desktop integrierten – Möglichkeiten zum gemeinsamen Lernen und Spielen können direkt genutzt werden.

Das OLPC-Projekt ist ein weltweites Projekt, das zurzeit mit über 60 sogenannten *Language Packs* die Sprachen und Tastatur-Layouts vieler Länder unterstützt. Um die XO-LiveCD ebenfalls weltweit verwenden zu können sind Optionen vorhanden, um zusätzliche Sprach-Pakete zu installieren und über Boot-Optionen zu aktivieren.

Das auf der LiveCD verwendete LiveBackup Framework [05] bietet darüber hinaus weitere Möglichkeiten bei einer Festplatteninstallation an. Auch ein Starten des Systems über Netzwerk (PXE-Boot) wird unterstützt. Sind auf einem Computer entsprechende Hardwarevoraussetzungen gegeben, kann der Systemstart auch von USB-Memory-Sticks oder SD/MMC-Speicherkarten erfolgen.

Das technische Prinzip

Auf dem XO-Laptop laufen SUGAR-Desktop und Activities unter einer angepassten *Red Hat* LINUX-Distribution. Der LINUX-Betriebssystem-Kern ist hier mit seinen Treibern speziell für die Hardware des XO-Laptops ausgelegt.

Auf der LiveCD wird das File-System als komprimiertes Abbild unverändert übernommen. Damit die Software auf möglichst vielen Systemen lauffähig wird, muss der Betriebssystem-Kern ausgetauscht und das System durch Hardware-Erkennung und Konfigurationen anpassungsfähig gemacht werden.

Neue Möglichkeiten eröffnen

Das Ziel der gemeinnützigen Stiftung *One Laptop Per Child* und aller Organisationen, die sich weltweit um das OLPC-Projekt gebildet haben, ist die Entwicklung eines kostengünstigen Laptops („100-$-Laptop“) – einer Technologie, die die Art und Weise, wie wir alle Kinder dieser Welt unterrichten, revo-

lutionieren könnte. Ihr Ziel ist es, allen Kindern neue Möglichkeiten zu eröffnen, zu forschen, zu experimentieren und sich auszudrücken.

Softwareentwicklung nach dem Open-Source-Modell setzt Engagement und auch Publizität des Projekts voraus. Gefragt sind IT-Experten genauso wie Pädagogen. Helfen Sie mit, indem Sie die Verbreitung der Ideen und Konzepte fördern! Ein erster Schritt könnte das gemeinsame Lernen und Spielen von Eltern und Kindern mit der XO-LiveCD sein.

Dr.-Ing. Wolfgang Rohrmoser
E-Mail:
contact@rohrmoser-engineering.de

Literatur

Freudenberg, R.: SUGAR – ein Betriebssystem zum Lernen. In: LOG IN, 29. Jg. (2009), Heft 156, S. 40–44 *(in diesem Heft)*.

Hinz, V.: Der XO-Laptop im Netz. In: LOG IN, 29. Jg. (2009), Heft 156, S. 65–70 *(in diesem Heft)*.

Internetquellen

[01] http://www.laptop.org/en/vision/project/index.shtml
[02] http://dev.laptop.org/pub/livebackupcd
[03] http://wiki.laptop.org/go/Activities
[04] http://www.quantatw.com/Quanta/english/product/qci_olpc.aspx
[05] http://livebackup.sourceforge.net/
[06] http://ted.streamguys.net/ted_negroponte_n_2006.mp3
[07] http://www.olpcnews.com/
[08] http://wiki.laptop.org/go/OLPC_Human_Interface_Guidelines
[09] http://www.opensource.org/
[10] http://olpcnepal.org/
[11] http://wiki.laptop.org/
[12] http://www.rohrmoser-engineering.de/
[13] http://beat.doebe.li/bibliothek/w00561.html
[14] http://scratch.mit.edu/
[15] http://www.squeakland.org/
[16] http://wiki.laptop.org/go/TamTam
[17] http://gcompris.net/
[18] http://lists.laptop.org/listinfo/livebackup-xo-cd
[19] http://wiki.laptop.org/go/Regional_community_groups
[20] http://news.squeak.org/2006/11/17/squeak-in-extremadura/
[21] http://wiki.laptop.org/go/Official_OLPC_FAQ#Who_is_behind_these_XO_laptop_computers.3F
[22] http://wiki.sugarlabs.org/go/Sugar_on_a_Stick

Alle Internetquellen wurden zuletzt am 20. März 2009 geprüft.

Hardware & Software

Digitale Fenster öffnen

Wie interaktive Tafeln Unterricht und Lernkultur einer Schule verändern können

Teil 1

Hohe Erwartungen an ein neues Medium

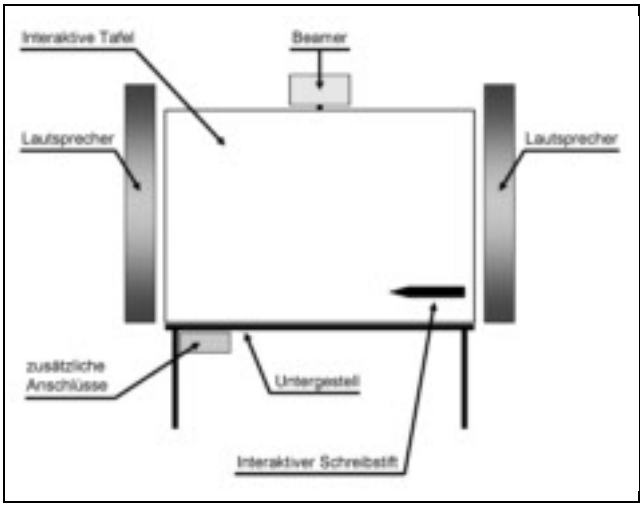
Möglicherweise markiert die diesjährige *didacta* den Durchbruch eines Unterrichtsmediums, das es zwar schon seit einigen Jahren gibt, welches aber bisher in deutschen Klassenzimmern noch keine große Bedeutung hatte. Sogenannte *interaktive Whiteboards* erobern mit Vehemenz die Schulen, und dieser Trend ließ sich in diesem Jahr während der Bildungsmesse in Hannover erkennen, wo interaktive Whiteboards allerorten zu finden waren (siehe auch Seite 4 f. in diesem Heft). Nicht nur die Hersteller von SMART Boards, Promethean Activboards, Interwrite Schoolboards, CleverBOARDs und etliche andere meldeten gestiegenes Interesse und hatten ihre Präsenz auf der Messe zum Teil erheblich erweitert, auch die großen und kleinen Bildungsverlage setzen nun deutlich auf den Trend zur digitalen Tafel und bieten Materialien für den Unterricht an. Versprochen wird „spannender Unterricht mit aktueller Zukunftstechnologie“, der durch die motivierenden „digitale[n] Alleskönner“ verwirklicht werden soll. Angefacht wird die Begeisterung für dieses Medium durch staatliche Gelder. In einigen Bundesländern gibt es seit längerem Investitionsoffensiven, die den Schulen Ressourcen speziell für interaktive Medien zur Verfügung stellen. Bereits 2007 beschloss beispielsweise Hamburg mit einem mehr als 20 Millionen Euro umfassenden Sonderinvestitionsprogramm, bis 2010 alle staatlichen Schulen breitbandig zu vernetzen und mit interaktiven Medien auszustatten. Der Stadtstaat setzt dabei Maßstäbe in der bundesrepublikanischen Bildungslandschaft, die nun auch andere Bundesländer wie Thüringen, Niedersachsen und Hessen mit Investitionen in diese Technologie erreichen wollen. Und auch das kürzlich beschlossene Konjunkturpaket II der Bundesregierung, das Schwerpunkte im Bereich Schulbau setzt, befördert das Interesse der Schulträger an digitalen Tafeln. Ziel ist dabei die Unterrichtsentwicklung und damit die Steigerung der Qualität von Schulen.

Hohe Erwartungen werden von vielen Seiten an das neue Unterrichtsmedium herangetragen – eigentlich überraschend, denn zunächst handelt es sich bei interaktiven Whiteboards um frontal im Klassenraum angebrachte elektronische *Tafeln*, den Inbegriff eines zentrierenden Mediums. Und von einem solchen Medium wird schon lange nicht mehr erwartet, dass es dazu beitragen könnte, den Unterricht im Hinblick auf individualisiertes und schüleraktivierendes Lernen zu modernisieren.

Die wenigen Studien und Untersuchungen, die bislang zum Einsatz interaktiver Whiteboards in Schulen und Unterricht vorliegen, geben kein klares Bild hinsichtlich des Mehrwerts dieses Mediums. Nach einer Berliner Studie von 2005 wird der Unterricht nach Meinung befragter Lehrkräfte interessanter, anschaulicher und verständlicher, wenngleich hervorgehoben wird, dass ein Mehraufwand in der Vorbereitung dafür in Kauf zu nehmen sei. Eine andere Untersuchung aus Hamburger Pilotschulen hebt vor allem den Motivationsfaktor hervor, der Schüler und Lehrer gleichermaßen im Unterricht beflügelt, wenn sie mit interaktiven Whiteboards arbeiten und lernen. Daneben gibt es nicht-repräsentative Hinweise auf die Gefahr der möglichen Zunahme des Frontalunterrichts durch den Einsatz solcher interaktiver Whiteboards. So weist ein Unterrichtsversuch aus Esslingen auf diese Gefahren hin. Alle Untersuchungen formulieren den Bedarf nach intensiver Lehrerfortbildung sowie die Notwendigkeit

Was ist ein interaktives Whiteboard?

Ein interaktives Whiteboard ist eine Weißtafel, die über USB oder Bluetooth mit einem Rechner verbunden wird. Dessen Bildsignale beleuchten über einen angeschlossenen Beamer unmittelbar das Whiteboard. Dadurch entsteht der Eindruck eines zeitgleichen realen Tafelbildes. Das Whiteboard fungiert im Grunde lediglich als Signaleingabegerät, das je nach Fabrikat mit dem Finger oder einem speziellen Stift bedient oder beschrieben werden kann. Das dadurch entstehende Tafelbild entsteht so in virtueller Form auf dem Rechner und kann deshalb digital abgespeichert und später wieder aufgerufen werden. Digitalisierte audio-visuelle Materialien können beim interaktiven Whiteboard leicht in Tafelbilder integriert werden. Für viele interaktive Whiteboards gibt es neben der notwendigen Treibersoftware auch pädagogische Lernsoftware für die Schule, die die Möglichkeiten der Tafelbild- und Unterrichtsgestaltung ausschöpfen soll.



Grundprinzip des Aufbaus einer interaktiven Tafel.

umfassender konzeptioneller Einbindung der interaktiven Whiteboards in die Schulentwicklung als Gelingensbedingungen. Insellösungen in vereinzelten Klassenzimmern werden als eher unvorteilhaft und wenig nachhaltig beschrieben.

Als Systemadministrator und Medienberater eines kleinen Hamburger Gymnasiums, das bis vor Kurzem nicht unbedingt als Vorreiter im Bereich *Lernen mit Neuen Medien* galt, habe ich erlebt, wie eine Schule von der Faszination der digitalen Tafel ergriffen wurde und sich in weniger als vier Jahren zu einer vollausgestatteten Schule mit interaktiven Whiteboards in allen Klassenräumen entwickelt hat. Dieser Prozess war begleitet von einem intensiven Diskurs über den Nutzen des Mediums sowie von didaktisch-methodischen und schulentwicklungsspezifischen Überlegungen zur Integration der Whiteboards und mündete in ein Integrationskonzept, das offensichtliche und schlummernde Potenziale dieses „Supermediums“ für den Unterricht und das Lernen an dieser Schule erschließen und dabei den möglichen Gefahren einer stärkeren „Frontalisierung“ des Unterrichts begegnen will. In dieser Beitragsreihe möchte ich Erfahrungen und Entscheidungen vorstellen und so einen möglichen Weg zum sinnvollen Unterricht und Lernen mit interaktiven Whiteboards skizzieren.

Begeisterung angesichts neuer Möglichkeiten im Unterricht

Erste Erfahrungen mit interaktiven Whiteboards sammelte unsere Schule, ein kleines humanistisches Gymnasium mit Musikzweig im Herzen Hamburgs, bereits vier Jahre vor der Ausstattung sämtlicher Klassenräume. Diese Vorlaufszeit erwies sich als wertvolle Vorbereitung und wichtige Voraussetzung für den großen Schritt der Vollausstattung. Im Rahmen eines Public-Private-Partnership-Projekts konnten wir im Jahr 2005 vier Boards anschaffen und diese erproben. Aufgrund des hohen Wertes der Geräte und der Gefahr des unsachgemäßen Umgangs damit wurden sie nach kurzer Überlegung in den besser gesicherten und nach dem Unterricht stets verschlossenen Fach- und Oberstufenräumen installiert. Dies hatte jedoch den Nachteil, dass nur wenige Lehrende und Lerngruppen die digitalen Tafeln unkompliziert und regelmäßig nutzen konnten, da man sich stets in die entsprechenden Räume einbuchen musste, was im Allgemeinen einen Raumwechsel voraussetzte. Nichtsdestoweniger weitete sich das Interesse an dem Medium innerhalb der Schule sehr schnell aus. Innerhalb kurzer Zeit hatten sich fast zwei Drittel des Kollegiums mit der Handhabung vertraut gemacht. Dies lag vor allem daran, dass schulinterne Fortbildungen am Nachmittag angeboten und vom Kollegium mit Interesse angenommen wurden. Parallel dazu war eine Keimzelle von Enthusiasten entstanden, die meist das Glück hatten, in ihrem Fachunterricht fest installierte interaktive Tafeln nutzen zu können. Sie wirkten als Multiplikatoren, die von den Vorzügen schwärmten und andere Kollegen anstifteten, das Medium zu erproben. Nach zwei Jahren konnte eine erste positive Bilanz gezogen werden: Schüler und Lehrer, die mit den Whiteboards arbeiteten und lernten, empfanden dies als hoch motivierend und wünschten sich mit deutlichen Mehrheiten die Ausweitung des Einsatzes auf andere Fächer und Räume. Der Unterricht wurde als verständlicher und anschaulicher eingeschätzt. Interessanterweise hatten die Schüler auch den Eindruck, dass stärker auf ihre Bedürfnisse eingegangen wurde als im Unterricht ohne interaktive Whiteboards.

Im Schulalltag entstand eine große Nachfrage nach Whiteboards,

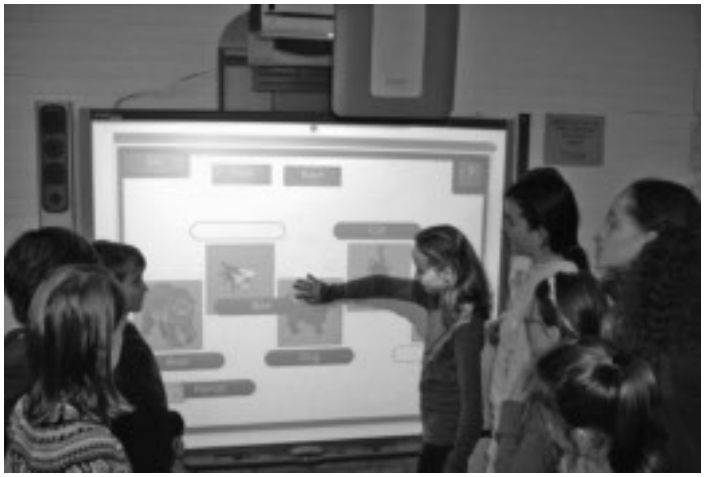


Foto: Wolfgang Thaler

Schüler-orientiertes Arbeiten am interaktiven Whiteboard.

die nicht mit der Anschaffung von ein, zwei Einheiten pro Jahr hätte befriedigt werden können. Gegen eine derartige Ausweitung der Medienausstattung sprach auch die Tatsache, dass bei einer Vielzahl verschiedener Modelle im Haus deren Wartung zu aufwendig geworden wäre. Aus dieser Situation heraus und aufgrund der Möglichkeit, staatliche Zuschüsse zur Anschaffung interaktiver Medien zu beantragen, diskutierte unsere Schule also nach drei Jahren die Frage, ob es nicht sinnvoll wäre, *alle* Klassenzimmer mit digitalen Tafeln auszustatten. Die Vorteile einer einheitlichen Lösung im Sinne von „Alle ans Board" lagen dabei auf der Hand:

- ▷ Es würde keine Zwei-Klassengesellschaft von Lehrenden und Klassen, die mit interaktiven Boards lernen, und jenen, denen dies nicht möglich ist, entstehen.
- ▷ Auch die Entscheidung, welche Klassen Anspruch auf die Tafeln haben würden, entfiele.
- ▷ Durch die Vollausstattung würden Redundanzen entstehen; aufgrund der Vielzahl der ausgestatteten Räume stünden im Falle von Störungen andere Geräte zur Verfügung.

Welche inhaltlichen Gründe sprachen für eine derart aufwendige Ausstattung? Zunächst sahen die meisten Kollegen den Nutzen der interaktiven Whiteboards auf der methodischen Ebene. Die deutlich verbesserten Gestaltungsmöglichkeiten für die Lehrenden waren ein wesentlicher Gesichtspunkt, der das Kollegium begeisterte:

- ▷ Die digitale Tafel integriert alle bisher genutzten Medien (Tafel, OH-Projektor, CD- und Video-Rekorder) in einem Gerät, das in jedem Klassenraum ohne Vorbereitung oder Raumwechsel zur Verfügung steht. Der Einsatz von audio-visuellen Medien wird dadurch erheblich erleichtert. Auch kurze Medieneinspielungen können im Unterricht zur Unterstützung von Lernprozessen herangezogen werden, da Aufwand und Nutzen in einem guten Verhältnis stehen. Dadurch können Lehrende und Lernende Präsentationsphasen besser medial unterstützen, und die Vermittlung von Inhalten wird verbessert.
- ▷ Die interaktiven Whiteboards integrieren aber nicht nur die oben genannten Medien, sondern vergrößern deren Potenzial vor allem im Bereich der Visualisierung erheblich. Hier liegt die große methodische Stärke der digitalen Tafel. Ihre speziellen Möglichkeiten erlauben die prozessorientierte Gestaltung „dynamischer Tafelbilder", die *gemeinsam mit den Schülern* entwickelt werden können. So können zum Beispiel Mindmaps im Unterricht zusammen erstellt werden, ohne dass Einschränkungen in ihrer Übersichtlichkeit hingenommen werden müssen. Hinsichtlich der gemeinsamen Strukturierung und (Re-)Organisation von inhaltlichen Zusammenhängen bietet dieses Medium ebenso enorme Potenziale.
- ▷ Das Abspeichern von Tafelbildern ermöglicht das Wiederholen und Anknüpfen an Gelerntes und schafft mehr Flexibilität im Unterricht. In den höheren Klassen erweist sich die digitale Verfügbarkeit der Tafelbilder als Vorteil; das Abschreiben von langen Anschrieben entfällt, die Unterrichtszeit kann effizienter genutzt werden.

Durch diese methodischen Vorteile der interaktiven Whiteboards wird der Unterricht nicht nur medienreicher und damit für mehr Lernertypen ansprechender, sondern durch dynamische, speicherbare Tafelbilder auch schülerorientierter und effektiver.

Chancen für neues Lernen

Neben diesen leicht nachvollziehbaren Vorteilen bieten die interaktiven Whiteboards jedoch auch eine herausragende Chance auf einer viel höheren Ebene. Das Lernen mit digitalen Medien spielte an unserer Schule eine Rolle, war aber zu jener Zeit noch nicht wirklich bei allen Kollegen und in allen Fächer verankert. Mit der Installation interaktiver Whiteboards *in allen Klassenräumen* rückte das Lernen mit neuen Medien nun ins Zentrum des Schulalltags. Bisherige Bemühungen der Schulentwicklung im Bereich der Medienkompetenz schufen viele Möglichkeiten für Lehrende und Lernende, die aber häufig nicht den Schulalltag charakterisierten, sondern eher die Ausnahme blieben; mit interaktiven Whiteboards im Klassenzimmer ändert sich dieser Zustand grundlegend. Das Angebot, digitale Medien für Lernprozesse zu nutzen, steht nun appellativ mitten im Klassenraum. Der Unterricht und das Lernen können durch diese Situation positiv verändert werden. Aber wie kann dieses Veränderungspotenzial aktiviert werden? Wie kann gewährleistet werden, dass es nicht nur bei anschaulicheren Tafelbildern und einem stärkeren Einsatz von audio-visuellen Medien im Unterricht bleibt, sondern auch weitere Schritte gegangen werden?

Entscheidend ist ein Medien- und Methodenkonzept der Schule, das eine veränderte Lernkultur im Sinne eines individualisierten Lernens fördert. Und hier eröffnen in-

teraktive Whiteboards enorme Chancen. Sie haben das Potenzial, das Lernen mit IuK-Techniken und Web 2.0 an einer Schule *insgesamt* voranzubringen, denn mit ihnen wird gleichsam ein digitales Fenster im herkömmlichen Klassenzimmer geöffnet, das es vorher in dieser Form in Schulen nicht gab. Lernen mit digitalen Medien war an den meisten Schulen – abgesehen von vereinzelten Notebookklassen – die Ausnahme. Meist schon aus räumlichen Gründen war langfristige Planung für deren Einsatz notwendig. Die digitale Tafel im Klassenzimmer kann diese Art des Lernens zum Normalfall werden lassen. Und sie bietet darüber hinaus den Vorteil, dass sich Schüler und Lehrer *gemeinsam* mit digitalen Lernumgebungen auseinandersetzen und so *voneinander* lernen können. Auch wird die Anleitung eines medienkompetenten Lehrers Schülern viele frustrierende Selbsterfahrungen ersparen und dadurch die Arbeit effektiver werden lassen. Lernen mithilfe digitaler Medien wird wesentlich kalkulierbarer und dadurch für viele Kollegen erst unterrichtstauglich. Zudem ermöglichen die interaktiven Whiteboards unerfahrenen Kollegen einen niederschwelligen Einstieg in die Arbeit mit digitalen Medien. Ein Lehrer, der sich vorher nie in den Computerraum verirrt hat, wird vielleicht damit beginnen, Tafelbilder digital zu erstellen und diese abzuspeichern, um sie in der nächsten Stunde noch einmal aufzurufen. Dabei erweist es sich als großer Vorteil, dass das interaktive Whiteboard zunächst als „klassisches“ Unterrichtsmedium daherkommt, als Tafel eben, die dem Kollegen vertraut ist. Möglicherweise stellt er im nächsten Schritt die Tafelbilder den Schülern digital zur Verfügung. Der entscheidende Schritt zur Öffnung des Unterrichts ist damit aber bereits getan, vor allem, wenn man bedenkt, dass die Schüler hier als Multiplikatoren sehr schnell darauf wirken dürften, den Austausch der Materialien jetzt auch noch online über einen virtuellen Lernraum erfolgen zu lassen. Digitale Medien können so auch jene Kollegen zu einer neuen, offeneren Lernkultur verführen, die sich vorher nicht auf derartige Lernarrangements eingelassen hätten; und mit den interaktiven Tafeln holen wir diese „digitale Verführung“ ins Zentrum des Unterrichts, mitten in den Klassenraum. Insofern haben die Boards das Potenzial, Unterricht und Lernkultur einer Schule von innen heraus grundsätzlich zu verändern.

Chancen für neues Lernen.

Foto: Wolfgang Thaler

Bedenken gegenüber der „Supertafel“

Auf der anderen Seite gab es während des Entscheidungsprozesses an unserer Schule aber auch Bedenken gegenüber der Einführung der interaktiven Whiteboards. Man befürchtete, dass der Einsatz der digitalen Tafeln negative Auswirkungen haben könnte:

- ▷ Die Energie- und Wartungskosten schienen für viele unabsehbar zu sein, wenn in allen Räumen nur noch mit Beamer und PC gearbeitet werden würde.
- ▷ Die Augen der Schüler könnten Schaden nehmen, wenn sie den ganzen Tag dem Beamerlicht ausgesetzt wären.
- ▷ Probleme mit Technik und Software würden wertvolle Unterrichtszeit kosten.
- ▷ Die Kompetenz der Lehrenden könnte für den Umgang mit digitalen Tafeln nicht ausreichen.
- ▷ Eine Zunahme des Frontalunterrichts als unmittelbare Folge der Existenz eines „Supermediums“ im Klassenraum wäre zu erwarten, was dem Bestreben nach einem verstärkt individualisierten Lernen entgegenstünde.

Die wenigen Untersuchungen zum Thema *Interaktive Whiteboards in Schulen* konnten uns nicht weiterhelfen, da sie kaum aussagekräftige Erkenntnisse speziell für unsere Fragen lieferten und sich vor allem stets auf Insellösungen bezogen; eine gesamtausgestattete Schule wurde bislang in Deutschland noch nicht untersucht. Diese Unwägbarkeiten veranlassten uns, ein Integrationskonzept zu erstellen, mit dem die Bedenken aufgegriffen, den möglichen Gefahren entgegengewirkt und die Stärken des Mediums gefördert werden sollten. Besonders der möglichen Verstärkung des Frontalunterrichts sollte entgegengewirkt werden, am besten mit der sinnvollen didaktischen Einbettung der interaktiven Whiteboards in ein modernes Medien- und Methodenkonzept der Schule. Auf dieser Grundlage entschied sich das Kollegium mit über 82 Prozent für die Vollausstattung; die anderen Schulgremien befürworteten die Entscheidung sogar mit noch größeren Mehrheiten.

(wird fortgesetzt)

Thomas Iser
E-Mail: mail@tiser.info

Aus redaktionellen Gründen (Platzmangel im vorliegenden Heft) erscheint der 3. Teil der Beitragsreihe von „Podcasts im Unterricht“ im nächsten Heft.

Rezension

Kleuker, Stephan: Grundkurs Software-Engineering mit UML – Der pragmatische Weg zu erfolgreichen Softwareprojekten. Wiesbaden: Vieweg & Teubner, 2009. ISBN 978-3-8348-0391-7. 361 S.; 29,90 EUR.

Die Softwaretechnik ist, in didaktisch reduzierter Form (etwa unter der Bezeichnung „Entwurf und Gestaltung von Informatiksystemen“) zentraler Teil eines bildungsstandardsgemäßen Informatikunterrichts. „Modellierung und Implementierung durchdringen alle Inhalts- und Prozessbereiche der informatischen Bildung“ (AKBSI, 2008, S. 45). Aus diesem Grund kann man an ein Buch, das beansprucht, „Lösungsansätze für die unterschiedlichen Herausforderungen des Software-Engineerings zu zeigen“ (Seite V), hohe Erwartungen richten: Verspricht es doch, das benötigte fachliche Hintergrundwissen bereitzustellen.

Die fachwissenschaftliche Grundlage des didaktischen Fünf-Phasen-Modells (Koerber/Peters, 1993) findet sich bei Kleuker in folgender Gestalt: *Anforderungsanalyse – Grobentwurf – Feinentwurf – Implementierung, Test und Integration* (alles begleitet von Maßnahmen zur Qualitätssicherung). Davon ausgehend diskutiert er die verschiedenen Vorgehensweisen, wie jene Phasen durchlaufen werden können: „Wasserfallmodell“, „V-Modell“, agile Vorgehensweisen, Scrum (siehe auch Steinbrucker, 2008, S. 5) usw.

Zu Beginn eines Projekts stellt sich immer wieder das Problem, wie die Anforderungen (engl.: *requirements*) an das zu erstellende System eindeutig und vollständig zu beschreiben seien. Diesem wichtigen Thema widmet Kleuker (in Kapitel 4) viel Aufmerksamkeit. Zwei zentrale Fragen sucht er zu beantworten: (1) Die Wünsche welcher Personen sind zu berücksichtigen? (2) Welche Ziele soll das Softwareprojekt erreichen? Die unterschiedlichen Perspektiven der Beteiligten (oder „Betroffenen“, engl.: *stakeholder*), also der Kunden, Manager, Entwickler, Tester usw. können zur Quelle vieler Missverständnisse werden; um sie möglichst zu vermeiden, müssen IT-Experten mit den Experten anderer Gebiete einvernehmlich zusammenarbeiten.

Die Hauptfunktionalitäten des Systems werden mittels sogenannter Anwendungsfälle (engl.: *use cases*) auf hohem Abstraktionsniveau modelliert. Nur solche Fälle werden ausgewählt, die außenstehende Anwender (Repräsentanten der *stakeholder*) wahrnehmen können. Mithilfe von Aktivitätsdiagrammen werden typische Abläufe festgehalten. Funktionale Anforderungen können mittels der Schablonen nach Rupp (vgl. auch Damm, 2001, S. 52) formuliert werden. Ziel des Grobentwurfs (Kapitel 5) ist es, die Aussagen der Anforderungsanalyse in die Sprache der Software-Entwickler zu übertragen. Hierzu eignen sich die Klassendiagramme der UML. Es wird gezeigt, wie man aus dem Anforderungstext zu einem ersten Klassendiagramm gelangt, und wie garantiert werden kann, dass letzteres die Realisierung der gewünschten Funktionalitäten ermöglicht.

Die folgenden Kapitel (6 bis 11) gehen auf unterschiedliche Aspekte des Wegs „Vom Klassendiagramm zum lauffähigen Programm“ ein. Es werden Techniken der objektorientierten Programmierung (in JAVA) sowie UML-Werkzeuge (Sequenzdiagramme usw.) behandelt. Um die Konzepte zu erläutern, weicht Kleuker auf gängige Lehrbuch-Beispiele (z. B. geometrische Probleme, Spiele) aus, statt beim Thema *Projektentwicklung* zu bleiben. Dies erweckt den Eindruck, als ob er hier lediglich Lehrbuchwissen, d. h. nicht eigenes Erfahrungswissen als Systemanalytiker und -berater wiedergibt. Zum Thema Qualitätssicherung finden sich nichtssagende Sätze wie: „Zentrale Aussage dieses Kapitels ist, dass man mit Test nur nachweisen kann, dass diese Tests in gewünschter Form ablaufen können“ (S. 245). Oder: „Die natürlichste Idee, wenn man zuerst über Korrektheit nachdenkt, ist der Ansatz, ein Programm zu schreiben, dem man das zu überprüfende Programm und eine Anforderung übergibt, das dann als Ergebnis die Antwort liefert, ob die Anforderung erfüllt ist“ (S. 246). Mit einem Verweis auf das sogenannte Halteproblem wird dann diese „natürliche Idee“ schnell fallen gelassen.

Fazit: Das Buch spricht alle die Themen an, die im Verlauf eines Softwareprojekts relevant werden können, und erörtert damit den fachlichen Hintergrund zentraler Teile des Informatikunterrichts. Es ist vor allem in den Anfangskapiteln, wo der Autor auf eigene Praxiserfahrungen zurückgreifen konnte, ausführlich und informativ. Im Mittelteil, bei Feinentwurf und Implementierung dagegen, ist es zu kursorisch und manchmal auch fachlich nicht immer ganz einwandfrei. Hier sollte man andere Werke (zu „Algorithmen und Datenstrukturen“) heranziehen. Dies gilt auch für die UML, von der aber immerhin gezeigt wird, wie sie in der Praxis eingesetzt wird. Das Buch ist kein „Meilenstein in der systematischen Ausbildung von Software-Entwicklern“, wie der Autor im Vorwort (Seite V) sich selbstgefällig attestiert, und es erfüllt auch nicht ganz die oben angesprochenen hohen Erwartungen. Doch kann es den Informatik-Lehrenden immerhin nützliche Hinweise und Einblicke in die Praxis des Software-Engineering geben.

Rüdeger Baumann

Literatur

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.

Damm, W.: Requirements Days 2001 – Das Problem des verständlichen Beschreibens von Problemen. In: LOG IN, 21. Jg. (2001), H. 5/6, S. 49–53.

Koerber, B.; Peters, I.-R.: Planungsstrukturen bei größeren Unterrichtsvorhaben – Theorie und Praxis der Planung informatischer Bildungsprozesse bei komplexen Problemlösungen. In: LOG IN, 13. Jg. (1993), H. 6, S. 19–25.

Steinbrucker, Chr.: Was wirklich gebraucht wird – Wünsche und Anforderungen von Soft-

wareproduzenten hinsichtlich der Informatik-Kompetenzen ihrer derzeitigen und künftigen Mitarbeiter. In: LOG IN, 28. Jg. (2008), Heft 154/155, S. 4–6.

Hinweise auf Bücher

Schulbücher

Weise, B.; Freiheit, H.-J.: Serielle Schnittstelle / Erfassen von Signalverläufen. Reihe „IT-Labor“, Band 1. Stuttgart: Holland + Josenhans Verlag, 2009. ISBN 978-3-7782-6025-8. 24 S.; 4,80 EUR.

Dieses DIN-A4-große Heft ist der erste Band einer neuen Reihe, die für den IT-Unterricht in technisch orientierten Schulen und Berufsschulen gedacht ist. Grundsätzlich ist jede Ausgabe des „IT-Labors“ in einen theoretischen und einen praktischen Teil gegliedert und wurde für die handlungsorientierte Gestaltung des Unterrichts mit praxisnahen Themen entwickelt. Der erste Band enthält zwei Versuche: I. Untersuchung serieller Schnittstellen und Darstellung dynamischer Größen, II. Erfassen von Signalverläufen auf Datenleitungen. Die Versuche sind so aufgebaut, dass sie sowohl individuell als auch in Teamarbeit bearbeitet werden können. Die dazu notwendigen Arbeitsblätter sind ebenfalls im Band enthalten und können sofort im Unterricht eingesetzt werden. Die Versuche wurden zum Teil mit Unternehmen und Technischen Fach- bzw. Hochschulen abgestimmt und im Unterricht erprobt. Das im ersten Band enthaltene Material deckt pro Versuch drei bis vier Stunden Unterricht in einem IT-Labor ab.

koe

Computer-Knobelei

Das Pomeranzenproblem

Im Jahr 1974 wurde bei einem Baseball-Turnier auf dem Gelände der Universität von Georgia der bisherige Rekord von 714 sogenannter *home runs* auf 715 verbessert. Ein junger Mathematik-Dozent namens Carl Pomerance, der diesem denkwürdigen Ereignis beiwohnte, fragte sich, ob das Zahlenpaar (714, 715) mathematisch von irgendeinem Interesse sei. Unter Mathematikern, insbesondere Zahlentheoretikern, ist es ein beliebter Zeitvertreib, über Zahlen, die ihnen im Alltag begegnet sind, allerlei Merkwürdiges und Amüsantes zu entdecken und publik zu machen.

Der berühmteste Fall dieser Art ist der folgende: Den indischen Mathematiker Srinivasa Ramanujan (1887–1920, Bild 1) besuchte einst sein Gastgeber und Förderer, der britische Zahlentheoretiker G. H. Hardy im Krankenhaus. Das Taxi, das ihn hergebracht habe, bemerkte Hardy beiläufig, „trug die Nummer 1729 – eine uninteressante Zahl, hoffentlich ist das kein schlechtes Omen“. – „Mitnichten“, entgegnete Ramanujan, „die Zahl ist hochinteressant, denn sie ist die kleinste natürliche Zahl, die sich auf genau zwei unterschiedliche Weisen als Summe zweier Kuben darstellen lässt: $a^3 + b^3 = 1729 = c^3 + d^3$ – übrigens eine Aufgabe von Euler.“

Bild 1:
Srinivasa Ramanujan (1887–1920).
http://www.math.rochester.edu/

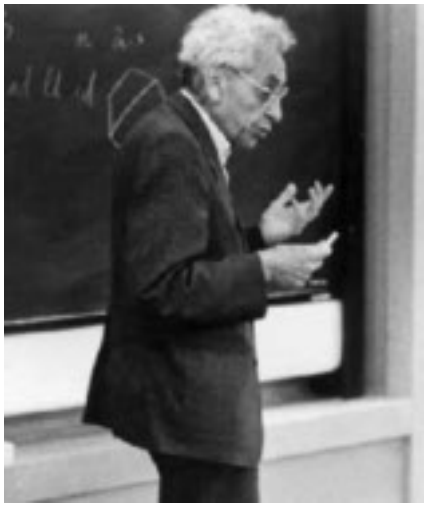
Diese Bemerkung entsprang jedoch keiner plötzlichen Eingebung Ramanujans; er hatte die arithmetische Kuriosität vielmehr früher einmal bemerkt und in seinem Notizbuch festgehalten. Aufgrund der für ihn charakteristischen Vertrautheit mit Zahlen hatte er sich dann an sie erinnert. Hardy verglich Ramanujan gern mit dem Berliner Mathematiker Carl Gustav Jacobi, auf dessen Scharfblick für Formeln und die Fähigkeit, Ergebnisse induktiv aus einer Vielzahl von Zahlenbeispielen zu gewinnen.

Aufgabe 1:
(a) Gesucht ist ein Programm, das Ramanujans Behauptung bestätigt und die beiden Zerlegungen angibt.
(b) Ermitteln Sie alle 52 Zerlegungen in je zwei Kuben (dritten Potenzen) bis 4 511 808 $= 100^3 + 152^3 = 27^3 + 165^3$.

Zurück zu Carl Pomerance. Der entdeckte, dass die Summen der Primfaktoren von $714 = 2 \cdot 3 \cdot 7 \cdot 17$ und $715 = 5 \cdot 11 \cdot 13$ miteinander übereinstimmen: es ist $2 + 3 + 7 + 17 = 29 = 5 + 11 + 13$. Das *Pomeranzenproblem* lautet: „Gibt es unendlich viele Zahlenpaare mit dieser Eigenschaft?“

Aufgabe 2: Schreiben Sie ein Programm, das die Anzahl aller Paare natürlicher Zahlen (n, n + 1) kleiner 100 000 mit der genannten Eigenschaft ermittelt. (Zur Kontrolle: unter 20 000 sind es 26, und zwar von n = 5 bis n = 18 490.)

Für die Zeitschrift *Journal of Recreational Mathematics* verfasste Pomerance einen launigen Artikel über seine Entdeckung. Obwohl sich die Häufigkeit der Zahlenpaare, genau wie die der Primzahlen, immer mehr verringert, nahm er an, dass es unendlich viele gäbe, hatte



http://www.cis.upenn.edu/

Bild 2: Paul Erdös (1913–1996).

aber keine Ahnung, wie sich seine Vermutung beweisen lasse. Schon in der ersten Woche nach Erscheinen des Aufsatzes bekam Pomerance einen Anruf von dem Mathematiker Paul Erdös (Bild 2), der ihm mitteilte, dass er das Problem gelöst habe, und um eine Einladung nach Atlanta bat, um den Beweis vortragen zu können. Pomerance war erstaunt und hoch erfreut, dass der berühmte Paul Erdös, dem er bisher noch nie begegnet war, von seiner Arbeit, die er als Scherz ohne Anspruch auf mathematischen „Tiefgang" ansah, überhaupt Notiz genommen hatte (Hoffmann, 1998, S. 223).

Paul Erdös (1913–1996) war einer der produktivsten und zugleich verschrobensten Mathematiker, die jemals gelebt haben. Etwa 1500 Abhandlungen (vornehmlich zur Zahlentheorie, Graphentheorie und abzählenden Kombinatorik) hat er, häufig mit Koautoren, verfasst. Erdös' Persönlichkeit erscheint als Inbegriff des „zerstreuten Professors": Er hatte keine Frau, keine Kinder, keinen Beruf, keine Liebhabereien, nicht einmal ein Zuhause, das ihn hätte binden können. Er lebte aus einem alten, schäbigen Koffer und einer abgegriffenen Plastiktüte. Auf der Suche nach ungelösten mathematischen Problemen und jungen Talenten reiste er von Konferenz zu Konferenz, von Institut zu Institut, quartierte sich bei Freunden und Bekannten ein und arbeitete sich mit Hilfe von reichlich Kaffee und Amphetaminen durch einen Neunzehnstundentag, den er ausschließlich der Mathematik widmete. „Ein Mathematiker", pflegte er zu sagen, „ist eine Maschine, die Kaffee in mathematische Sätze verwandelt." Da er im Wesentlichen auf Kosten seiner Gastgeber lebte, konnte er seine Honorare verschenken. Als Erdös einmal in Madras vortrug, sandte er sein Honorar an Ramanujans bedürftige Witwe; er hatte Ramanujan nie kennengelernt, war aber als junger Mathematiker von dessen Ideen inspiriert worden.

Zahllose Erdös-Anekdoten sind überliefert. Als dieser einmal im Rahmen einer Konferenz interviewt wurde, kam die Frage: „Wenn ich ihnen zwei vierstellige Zahlen nennen würde, könnten Sie sie im Kopf miteinander multiplizieren?" – „Heute nicht mehr", antwortete Erdös, „aber als ich vier war, konnte ich das!" Er litt unter der Vorstellung, mit fortschreitendem Alter „immer dümmer" zu werden, was er wiederum zum Anlass nahm, über sich selbst zu witzeln. Schon als Siebzehnjähriger äußerte er gegenüber einem Freund: „Das Quadrat von 2532 ist 6411024; leider werde ich alt und vertrottelt, darum kann ich dir die dritte Potenz nicht sagen." Auf seinem Grabstein sollte stehen: „Endlich kann ich nicht mehr dümmer werden."

Zuschriften bitte an:

Rüdeger Baumann
Fuchsgarten 3
30823 Garbsen

E-Mail:
baumann-garbsen@t-online.de

Literatur

Aigner, M.; Ziegler, G.M.: Das BUCH der Beweise. Berlin u. a.: Springer, ²2004.

Hoffman, P.: Der Mann, der nur Zahlen liebte. Berlin: Ullstein, 1999.

Veranstaltungskalender

27.–31. Juli 2009:
WCCE 2009 – 9th IFIP World Conference on Computers in Education
Fundaparque, Bento Gonçalves (RS), Brasilien

Information:
http://www.wcce2009.org/

14.–17. September 2009:
DeLFI 2009 – 7. E-Learning Fachtagung der Gesellschaft für Informatik
Freie Universität Berlin

Information:
http://www.e-learning2009.de/delfi/index.html

Die DeLFI-Tagung widmet sich allen Aspekten rechnergestützten Lernens und Lehrens. Ausgewählte Fragestellungen dieses Themenkomplexes werden durch Vorträge intensiv behandelt. Der Schwerpunkt liegt auf den Anforderungen an die Informatik und den daraus resultierenden Ergebnissen.

21.–24. September 2009:
INFOS 2009
Freie Universität Berlin

Information:
http://www.infos2009.de/

Die 13. GI-Fachtagung „Informatik und Schule" steht unter dem Motto „Zukunft braucht Herkunft – 25 Jahre INFOS". Näheres, insbesondere der „Call for Papers" ist in LOG IN 153 (2008), S. 84 f. zu finden.

28. September – 2. Oktober 2009:
INFORMATIK 2009 – 39. Jahrestagung der Gesellschaft für Informatik e. V. (GI)
Universität zu Lübeck

Information:
http://www.informatik2009.de/

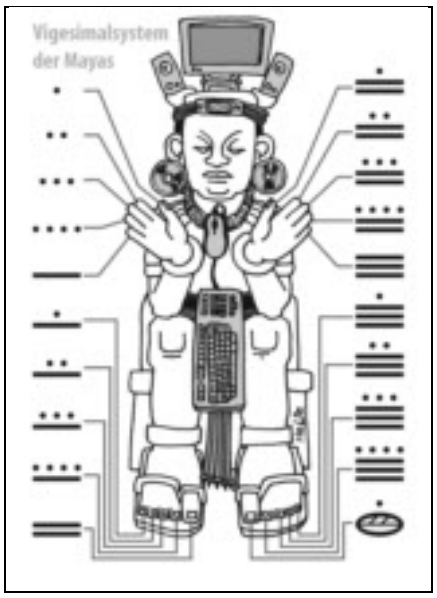
LOG OUT

Warum wir im Dezimalsystem rechnen

Die Überschrift könnte Verwirrung stiften, denn die Antwort scheint klar: Natürlich, weil wir 10 Finger haben. Dies provoziert jedoch eine Gegenfrage: Heißt das etwa, dass wir z. B. im 60er-Zahlensystem rechnen würden, wenn uns die Natur mit 60 Fingern ausgestattet hätte? Immerhin haben beispielsweise die Mayas im Vigesimalsystem (Zwanzigersystem) gerechnet, weil sie neben ihren Fingern auch die Zehen zum Zählen benutzten. Welchen Unterschied macht es eigentlich, ob wir im 2er-, 10er- oder 60er-Zahlensystem rechnen?

Schauen wir uns als Erstes die Länge der Zahlen an. Der Dezimalzahl 1024 entspricht die Dualzahl 10000000000. Die eine sprechen wir mit „eintausendvierundzwanzig“ aus, die andere konsequenterweise mit „zehnmilliarden“. Wunderschön auszusprechen ist die Dualzahl 1111101000: „einemilliardeeinhundertelfmillioneneinhunderteinstausend“. Die Staatsverschuldung in Deutschland betrug in dem Augenblick, als ich dies hier aufschreibe: 1 556 241 617 059 Euro (Quelle: http://www.steuerzahler.de/; Stand: 18. 04.2009). Die entsprechende Dualzahl hat 41 Stellen. Die eine kann ich noch aussprechen, die andere nicht mehr. Natürlich kann ich deren Ziffern aufsagen. Übersichtlich ist das jedoch nicht. Ziehen wir ein erstes Fazit: Wegen des Sprech- und Schreibaufwands sollte die Zahlenbasis groß sein.

Was muss man sich für das Rechnen mit den Zahlen merken? Mindestens die zugrunde liegenden Algorithmen, die Plustabelle und die Maltabelle. Die Algorithmen an sich hängen nicht von der Zahlenbasis ab. Wie schaut es mit der Maltabelle aus, dem Einmaleins? Bei der Zahlenbasis 10 muss man sich 36 relevante Tabellenfelder merken. (Die Ergebnisse von 0-mal irgendwas und 1-mal irgendwas bleiben unbeachtet; die Kommutativität der Multiplikation wird ausgenutzt.) Bei der Zahlenbasis 2 muss man sich nichts merken. Allgemein muss man sich $(z-2)(z-1)/2$ relevante Tabellenfelder merken (z ist die Zahlenbasis). Die Anzahl steigt also quadratisch. Bei der Zahlenbasis 60 sind das immerhin 1711 Tabellenfelder. Dieses Einmaleins wäre eine ungeheure Herausforderung. Ziehen wir ein zweites Fazit: Wegen des Merkaufwands sollte die Zahlenbasis klein sein.



Zeichnung: J.-H. Dahmen

Spricht noch etwas für eine große oder eine kleine Zahlenbasis? Bei kurzen Zahlen hat man beim Multiplizieren vergleichsweise wenige Teiloperationen auszuführen. Kurze Zahlen erhält man bei einer großen Zahlenbasis. Ziehen wir ein drittes Fazit: Wegen des Rechenaufwands sollte die Zahlenbasis groß sein.

Fassen wir zusammen: Manches spricht für eine große, manches für eine kleine Zahlenbasis. Ein sinnvoller Kompromiss unter besonderer Beachtung unserer Merkfähigkeit scheint für uns Menschen eine Zahlenbasis in der Nähe der 10 zu sein.

Und jetzt erst kommen unsere Finger ins Spiel.

Michael Fothe

Vorschau

Heft 157 – 29. Jg. (2009)

Thema: Informatikgeschichte im Informatikunterricht
Koordination: Marco Thomas

Thema von Heft 158:

▷ Präsentieren – Eine Kompetenz im Informatikunterricht

Thema von Heft 159:

▷ Veranschaulichung – Modelle und Realität

Mitarbeit der Leserinnen und Leser

Manuskripte von Leserinnen und Lesern sind willkommen und sind an die Redaktionsleitung in Berlin – am besten als Anhang per E-Mail – zu senden. Auch unverlangt eingesandte Manuskripte werden sorgfältig geprüft. Autorenhinweise werden auf Anforderung gern zugesandt.

LOG-IN-Service

Mit dem LOG-IN-Service bietet die Redaktion seit dem Heft 4/1991 regelmäßig Software, Unterrichtsmaterialien bzw. besondere Informationen kostenfrei für alle Abonnenten an.

Der LOG-IN-Service ist auf der Internetpräsenz des Verlags zu finden:

http://www.log-in-verlag.de/

Service zum Heft 156

Im LOG-IN-Service dieses Hefts sind verfügbar:

▷ Zum Beitrag „Ein Ausflug in den Compilerbau“ (S. 59–64) die vorgestellten JAVA-Programme und ein Ausschnitt aus der JAVA-Grammatik.

Freie Software und Bildung e.V.

Im Rahmen des Vereinszwecks fördert der FSuB auch die Verbreitung der OLPC (One Laptop per Child) Software.

Sie können auf Ihrem Computer die Software des OLPC ausprobieren, ohne dass der Rechner verändert werden muss, da das System komplett von der CD startet und läuft: Mesh Netzwerk (adhoc Netzwerk) ausprobieren, mit dem Netzwerk kommunizieren, Entwicklungsumgebung mit Phython ausprobieren Squeak (Programmierumgebung auch für jüngere Kinder) ausprobieren. Den Rest entdecken Ihre Schüler alleine.

Neben der OLPC-LiveCD fördert der FSuB Skolelinux, eine Initiative, die einen kompletten Schulserver anbietet. Diese Software findet weltweite Verbreitung und wurde zunächst in Norwegen entwickelt.

Den Verein „Freie Software und Bildung“ gibt es seit 1999. Der Verein setzte sich damals bei der Gründung zum Ziel, die Situation von Schulen und anderen nicht kommerziellen Bildungsträgern im Bereich der neuen Informations- und Kommunikationstechnologien zu verbessern sowie Informationen im Internet zugänglich zu machen, die es auch Einzelpersonen ermöglicht, sich in diesen Bereichen zu orientieren. Besonders möchte der FSuB die Verbreitung von freier offener Software (FoSS) vorantreiben. An der folgenden Analyse aus der Gründerzeit hat sich auch in der heutigen Schulsitutation, 10 Jahre später, immer noch nichts Grundlegendes geändert.

Analyse

Heute wie damals gilt die Analyse natürlich nur für den größten Teil der Bildungseinrichtungen, nicht für alle.

- Vielen Lehr- und Unterrichtskräften fehlt eine grundlegende Ausbildung im Sachbereich neue Informations- und Kommunikationstechnologien. Bei jungen LehrerInnen verbessert sich die Situation durch die Studienseminare deutlich.
- Die Verwaltung der schulischen Rechner liegt im Allgemeinen bei den Lehrkräften selbst, die meist für diese Aufgabe nicht ausgebildet wurden.
- Die für die Verwaltung der schulischen Rechner erforderliche Zeit wird den Lehrkräften nur zu einem relativ geringem Teil als Arbeitszeit zugestanden. Um den Rechnerbetrieb aufrechtzuerhalten, müssen große Teile der privaten, freien Zeit aufgewendet werden.
- Die für die Aus- und Fortbildung erforderliche Zeit, wird den Lehrkräften nicht gewährt – außer in der Freizeit!
- Die Unterrichtsinhalte sind nicht reflektiert: In weiten Teilen der Bundesrepublik wird überwiegend unterrichtet, wie die Anwendungsprogramme einer einzigen, weltweit dominanten Softwarefirma zu bedienen sind.
- Die Ausstattung mit Hardware ist oft mangelhaft.

Freie Software hat Vorteile

- Die in den Schulen eingesetzte Software kann von Lehrern und Schülern zu Hause lizenzkostenfrei eingesetzt werden. Damit kann gleichzeitig erreicht werden, dass
- der Unterricht zu Hause vor- und nachbereitet werden kann,
- keine Lehrer oder Schüler zum Raubkopieren verleitet werden,
- Lehrer nicht für etwaiges Raubkopieren in der Schule zur Verantwortung gezogen werden können.
- Der Einsatz einer freien und offenen Betriebssystemsoftware bietet auf der Angebotsseite europäischen Firmen die Chance, Ihr Know-how einzubringen und weiterzuentwickeln und sichert damit Arbeitsplätze in Europa. Das höhere Steueraufkommen könnte man den Schulen zugutekommen lassen. Auf der Nachfrageseite spart es Kosten und fördert damit ebenfalls die internationale Wettbewerbsfähigkeit.

Mitmachen

Wir suchen Mitstreiterinnen und Mitstreiter. Entweder als unterstützende Mitglieder, die durch ihren Mitgliedsbeitrag die Vereinsziele fördern oder als aktive Mitglieder, die mit überlegen, wie die Verwendung offener Software in der Bildung vorangetrieben werden kann.
Infos: http://www.fsub.de
Mitgliedsantrag: http://fsub.schule.de/fsub/fsub-antrag.pdf

www.cotec.de

Auf www.cotec.de finden Sie unser gesamtes Angebot an Software, Hardware, Videos und Büchern zu attraktiven Preisen, immer unsere neuesten Sonderangebote und den co.Tec Gruppenkauf - *gemeinsam sparen!* Lassen Sie sich einfach über unsere kostenlosen eNews informieren.

Jetzt kostenlos anfordern!

co.Tec GmbH
Traberhofstraße 12
83026 Rosenheim
Telefon: 08031 2635 - 0
Telefax: 08031 2635 - 29
www.cotec.de
info@cotec.de

Softwarepartner der Schulen