

FRIEDRICH-SCHILLER- UNIVERSITÄT JENA

Fakultät für Mathematik und Informatik



seit 1558

JENAER SCHRIFTEN ZUR MATHEMATIK UND INFORMATIK

Eingang: 20. Dezember 2004 Math / Inf / 13 / 04
Als Manuskript gedruckt

Unterricht – bald nur noch mit Computer?

Öffentliche Antrittsvorlesung am 8. Oktober 2004

Michael Fothe
Casio-Stiftungsprofessur
für Didaktik der Informatik/Mathematik
Friedrich-Schiller-Universität Jena
Fakultät für Mathematik und Informatik
07740 Jena
fothe@minet.uni-jena.de

Unterricht – bald nur noch mit Computer?

Öffentliche Antrittsvorlesung am 8. Oktober 2004

1. Einführung

Grundpositionen zum Thema stellt der Autor in drei Thesen vor. Dabei beantwortet er u. a. die Frage, ob die Schülerinnen und Schüler an den Schulen künftig von Computern unterrichtet werden sollten. Dann werden Möglichkeiten, Konsequenzen und Probleme des Computereinsatzes im Unterricht anhand von Beispielen aus Mathematik, Informatik und Chemie thematisiert. Als wesentlich wird die Ausgestaltung der Mensch-Maschine-Schnittstelle herausgestellt. Nachfolgend geht es auch um Veränderungen im Mathematikunterricht, um Ziele und Inhalte des Informatikunterrichts und um Kooperation von Mathematik- und Informatiklehrern an Schulen. Das Vorstellen von Aufgaben einer Fachdidaktik und von Forschungsvorhaben innerhalb der Casio-Stiftungsprofessur für Didaktik der Informatik/Mathematik rundet diese Ausarbeitung zur Antrittsvorlesung ab.

2. Zentrale Thesen

These 1: Der Einsatz von Computern im Unterricht bietet interessante und nützliche didaktisch-methodische Möglichkeiten.

Die Möglichkeiten gilt es für die einzelnen Unterrichtsfächer zu erschließen. Für den Einsatz von Medien und Werkzeugen im Unterricht sind Konzeptionen erforderlich. Diese sind rechtzeitig zu erarbeiten, um an den Schulen zu agieren und nicht nur zu reagieren. Eine wichtige Frage dabei ist, was mithilfe des Mediums oder Werkzeugs getan wird und was ohne Hilfsmittel zu erledigen ist.

In Thüringen gibt es mittlerweile 47 Medienschulen. Interviews mit ausgewählten Schülerinnen und Schülern der ersten fünf Medienschulen ergaben das folgende Bild: 75% der Schüler beantworteten die Frage, ob in ihrer Klasse die Lernmotivation durch den Einsatz neuer Medien zugenommen hat, mit Ja. 65% der Schüler sagten, dass die Leistungsbereitschaft der Schüler der eigenen Klasse beim Einsatz neuer Medien spürbar höher ist (vgl. [11]).

These 2: Computer werden immer mehr zu kognitiven Maschinen. Diese Tatsache wird sich als Chance und Herausforderung für die Schule erweisen.

Man denke an Textverarbeitungssysteme, von denen die Rechtschreibung recht gut und die Grammatik bereits ansatzweise überprüft werden. Solche Systeme sind auch hilfreich für Schülerinnen und Schüler. In manchen Branchen werden bereits Diktiersysteme mit Erfolg eingesetzt.

These 3: Computern, die irgendwann den Turingtest bestehen, wird man wohl Intelligenz zugestehen müssen. Menschengleich sind sie dann noch lange nicht. Daher sind Lehrerinnen und Lehrer nicht abschaffbar.

Bei der Nachahmung des Menschen im Sinne der künstlichen Intelligenz sind neben Intelligenz auch Bewusstsein und freier Wille zu verlangen (vgl. [3]). Folgerichtig wird nicht dazu aufgerufen, Lehrerinnen und Lehrer durch Computer zu ersetzen. Im Übrigen beträgt das Lehrer-Schüler-Verhältnis an den Thüringer Gymnasien 1 : 15,1 in den Klassenstufen 5-10 und 1 : 11 in den Klassenstufen 11-12. Das Computer-Schüler-Verhältnis ist 1 : 11. Schülerinnen und Schüler der 11. und 12. Klassen tref-

fen mittlerweile also an ihren Schulen auf genau so viele Computer wie auf Lehrerinnen und Lehrer.

3. Kommunikation zwischen Schüler und Computer

Bei der Entwicklung von pädagogischer Software ist besondere Sorgfalt auf das Ausgestalten der „Schüler-Computer-Schnittstelle“ zu legen, was der folgende Dialog deutlich macht:

- Computer: Wieviel ist $2 + 2$?
- Schüler: vier
- Computer: Falsch! Kann ich dir helfen?
- Schüler: 4
- Computer: Ich verstehe dich nicht.
- Schüler: Wenn du mich nicht verstehst, dann kann ich auch nicht mit dir lernen!

Für das Ausgestalten der Schnittstelle wird ein Beispiel aus dem Chemieunterricht angegeben: Zu erarbeiten ist ein Programm, das Schülerinnen und Schüler beim Erlernen der chemischen Zeichensprache hilft. Der einfache Vergleich zweier Zeichenfolgen – die korrekte, die im Programm gespeichert ist, und die vom Schüler eingegebene – wäre unzureichend (siehe die weitere Darstellung).

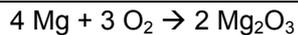
Zuerst wird die formale Sprache der Reaktionsgleichungen für Oxidations- und Redoxreaktionen mittels der erweiterten Backus-Naur-Form (EBNF) definiert:

```
Bestandteil1 = [ Faktor ] Elementsymbol.  
Bestandteil2 = [ Faktor ] O2.  
Bestandteil3 = [ Faktor ] Oxidformel.  
Oxidationsgleichung = (( Bestandteil1 "+" Bestandteil2 ) | ( Bestandteil2 "+" Bestandteil1 ))  
                    "+" Bestandteil3.  
Redoxgleichung = (( Bestandteil1 "+" Bestandteil3 ) | ( Bestandteil3 "+" Bestandteil1 )) "+"  
                (( Bestandteil3 "+" Bestandteil1 ) | ( Bestandteil1 "+" Bestandteil3 )).
```

Zusätzlich werden Kontextbedingungen in verbaler Form hinzugefügt (Symbole, Wertigkeiten, Ladungen usw.). Die Analyse einer Schülerantwort erfolgt in drei Schritten. Im ersten Schritt werden so genannte Eingabefehler erkannt und der Schüler erhält ggf. die Möglichkeit zur Korrektur seiner Zeichenfolge. Wurde die Definition der Eingabesyntax eingehalten, so erfolgt eine interne Übersetzung in die 1. Zwischensprache. Im zweiten Schritt wird überprüft, ob die Syntax der chemischen Zeichen und andere Kontextbedingungen erfüllt sind. Es erfolgt ein Übersetzen in die 2. Zwischensprache. Im dritten Schritt wird die Eingabe des Schülers mit der Vorgabelösung verglichen. Dabei wird versucht, die Schülerantwort nachzuvollziehen. Man spricht von generativen Antwortkontrollverfahren.

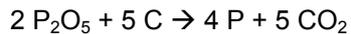
Einige Beispiele für Reaktionen des Programms:

```
3 HCl + Al(OH3) → AlCl3 + 3 H2O  
-----!  
Eingabefehler  
  
2 Cy + O2 → 2 CuO  
Cy ist ein unbekanntes Symbol.  
  
Mg + O → MgO  
Sauerstoff tritt als Molekül auf.
```



Die Oxidformel ist falsch.

Magnesium hat nicht die Wertigkeit 3.



Richtig.

Das Vorgehen lehnt sich an übliche Verfahren zur Syntaxanalyse an. Grundlagen für das Programm waren Erfahrungen des Programmautors und empirische Untersuchungen. Das Programm stellt insofern implementierte Lehr-Erfahrung dar. Der Computer kann zum geduldigen und kompetenten Überprüfer von Antworten der Schülerinnen und Schüler werden (vgl. [7]).

4. Rechnen als Kulturtechnik

Breite Volksschichten mussten in den letzten Jahrhunderten, vor allem aus wirtschaftlichen Gründen, das Rechnen erlernen. Dies gelang recht gut, indem Handlungen von deren Bedeutung entkoppelt wurden. Ein Beispiel ist die Multiplikation $4,8 * 26,01$. Viele können perfekt multiplizieren und wissen, dass die Teilprodukte versetzt aufzuschreiben sind. Warum das so ist, ist oft unklar. Im täglichen Leben nutzt mittlerweile fast jeder den Taschenrechner und macht sich damit von der Technik abhängig. Die nicht verstandenen Handlungen werden ruhigen Gewissens dem Taschenrechner übertragen. Das Standardbeispiel „Verkäuferin“ überzeugt die Schülerinnen und Schüler nicht mehr, das Rechnen ohne Hilfsmittel zu erlernen. Für den Einsatz von Taschenrechnern an Schulen können fünf Forderungen erhoben werden:

- Anerkennen der universellen Verfügbarkeit der Taschenrechner,
- Beenden des Einsatzes von Schülern als Prozessoren für das Abarbeiten von Algorithmen, die ein Taschenrechner ausführen kann,
- Nutzen verschiedener Taschenrechner-Typen im Verlaufe der Schulzeit,
- Vermitteln der Notwendigkeit, die vom Taschenrechner gelieferten Angaben kritisch zu bewerten, und
- Befähigen der Schülerinnen und Schüler zu einem grundlegenden Zahlenverständnis.

Ein Beispiel für grundlegendes Zahlenverständnis: $24 : 0,4$ müssen die Schülerinnen und Schüler im Kopf bewältigen. $23,41 : 0,387$ können sie dem Taschenrechner überlassen (vgl. [8]). Sicher nicht aus Versehen gab Gottfried Wilhelm Leibniz (1646-1716) seiner Rechenmaschine den Wahlspruch SUPRA HOMINEM. Sie ist also dem Menschen überlegen, was Leibniz damit begründete, dass sie den Menschen durch die Schnelligkeit und Zuverlässigkeit größter Berechnungen übertrifft¹.

5. Computer im Mathematikunterricht

Der Einsatz von Tabellenkalkulationen, Computeralgebrasystemen (CAS) und dynamischer Geometriesoftware wird aus didaktisch-methodischer Sicht für den Mathematikunterricht als sehr wünschenswert angesehen (vgl. [10]). Die Computer sollen dabei nicht als Selbstzweck genutzt werden. Vielmehr werden in ihnen Werkzeuge gesehen, mit deren Hilfe Mathematikunterricht verbessert werden kann. Die Notwendigkeit inhaltlicher Verbesserung ist u. a. durch das eher mittelmäßige Abschneiden deutscher Schülerinnen und Schüler bei internationalen Vergleichsuntersuchungen deutlich geworden. Ein wichtiges Unterrichtsziel, das nur langfristig zu erreichen ist,

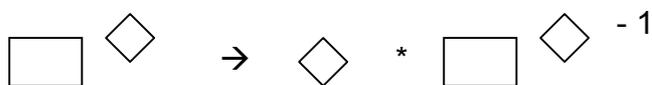
¹ Mehr zu G. W. Leibniz, seiner nicht vollständig funktionierenden Rechenmaschine und dem funktionierenden Nachbau von N. J. Lehmann (1921-1998) kann auf der Homepage der Casio-Stiftungsprofessur nachgelesen werden. Leibniz war zeitweise Student von Erhard Weigel (1625-1699) in Jena, der die Zahlbasis 4 vorschlug.

ist die Befähigung der Schülerinnen und Schüler zur selbstständigen Auswahl von Werkzeugen, die für eine bestimmte Problemlösung besonders geeignet sind. In den einheitlichen Prüfungsanforderungen der KMK in der Abiturprüfung Mathematik wird dargestellt, dass neue Technologien wirksam eingesetzt werden können bei dynamischen Visualisierungen zum Aufbau von Grundvorstellungen mathematischer Begriffe, als leistungsfähige Werkzeuge bei Modellbildungen und Simulationen sowie zur Förderung heuristisch-experimentellen Arbeitens (vgl. [5]).

Aus dem Alltag weiß man, dass es sinnvoll ist, Aufbau und Arbeitsweise von Werkzeugen zumindest im Prinzip zu kennen. Als Beispiel sei das Auto genannt. Folgerichtig sind Motor und Getriebe Gegenstand von Physik- oder Technikunterricht. CAS können symbolisch differenzieren. Die Zeichenfolge $7x^2 - 5x + 3$ überführen sie in $14x - 5$. Die Zeichenfolge $x^4 + \sin x * x^2 + 2$ wird zu $4x^3 + \sin x * 2 * x + \cos x * x^2$. Für das Werkzeug CAS soll die These formuliert werden, dass Transparenz z. B. durch Programmieren erreichbar ist. Die Kooperation von Mathematik- und Informatiklehrern einer Schule ist dabei sinnvoll. Im Mathematikunterricht werden die mathematischen Grundlagen erarbeitet (z. B. Ableitungsbegriff, Differenzierungsregeln, Ableitung spezieller Funktionen). Ein CAS wird im Unterricht vielfältig genutzt. Im Informatikunterricht wird ein Programm erarbeitet, das das symbolische Differenzieren ausführt. Eine geeignete Programmiersprache dafür ist Prolog. Prolog stellt Mechanismen zum automatischen Zerlegen von Zeichenfolgen in ihre Bestandteile unter Berücksichtigung der Vorrangregeln der Operatoren bereit (vgl. [2]). Wir verwenden das dreistellige Prädikat $d(F, X, DF)$. F ist die gegebene Zeichenfolge. DF liefert das Resultat, also die Zeichenfolge, die die 1. Ableitung darstellt. X ist eine Variable für die Variable, nach der differenziert wird. Für spezielle Funktionen werden die folgenden Fakten und Regeln angegeben:

```
d ( X , X , 1 ).
d ( C , X , 0 ) :- atomic ( C ) , C \= X.
d ( pot ( X , N ) , X , N * pot ( X , N1 ) ) :- N > 1 , N1 is N - 1.
d ( sin ( X ) , X , cos ( X ) ).
d ( cos ( X ) , X , - sin ( X ) ).
d ( exp ( X ) , X , exp ( X ) ).
d ( ln ( X ) , X , 1 / X ).
```

Zum Beispiel bedeutet die fünfte Zeile des Prolog-Programms: Die Zeichenfolge $\cos(X)$ wird in die Zeichenfolge $-\sin(X)$ überführt. Grafisch lässt sich der Ersetzungsprozess z. B. für die Potenzfunktionen in Anlehnung an die Symbolik bei formalen Sprachen wie folgt darstellen:



Summen- und Produktregel lassen sich in Prolog beschreiben:

```
d ( F + G , X , DF + DG ) :- d ( F , X , DF ) , d ( G , X , DG ).
d ( F * G , X , F * DG + DF * G ) :- d ( F , X , DF ) , d ( G , X , DG ).
```

Differenz- und Quotientenregel sind entsprechend zu formulieren.

Die Produktregel lässt sich visualisieren:



Weiß sind die Funktionen, schwarz deren 1. Ableitungen dargestellt.

Die Kettenregel wird für jede Grundfunktion einzeln angegeben:

$d(\sin(U), X, \cos(U) * DU) :- d(U, X, DU).$ $d(\text{pot}(U, N), X, N * \text{pot}(U, N1) * DU) :- N > 1, N1 \text{ is } N - 1, d(U, X, DU).$

DU ist die Zeichenfolge, die der inneren Ableitung entspricht.

Die Kettenregel wird grafisch dargestellt:



Das vollständige Programm besteht nur aus rund 20 Zeilen. Es ist durchaus leistungsfähig, denn sogar mehrfach verschachtelte Funktionen werden von ihm richtig bearbeitet. Die Korrektheit des Ergebnisses ist z. B. bei fünffacher Verschachtelung kaum noch zu überprüfen, da selbst ein geübter Mensch den Überblick über die äußeren und inneren Ableitungen verlieren dürfte. Aus gutem Grund stellen Lehrerinnen und Lehrer in der Schule meistens Aufgaben, bei denen sie die Antworten der Schülerinnen und Schüler korrigieren können.

6. Ziele und Inhalte des Informatikunterrichts

Als geeignete Grundlage für die Konzeption von Informatiklehrplänen und -unterricht haben sich die vier Leitlinien zur informatischen Bildung erwiesen: Interaktion mit Informatiksystemen, Wirkprinzipien von Informatiksystemen, informatische Modellierung sowie Wechselwirkungen zwischen Informatiksystemen, Individuum und Gesellschaft (vgl. [6]). Auf die Expertise zur Entwicklung nationaler Bildungsstandards soll zumindest hingewiesen werden. In ihr werden fünf Kulturwerkzeuge angegeben. Mathematisierungskompetenz und IT-Kompetenz gehören dazu (vgl. [12]). Zur Konsequenz, auch nationale Bildungsstandards für Informatik zu entwickeln, hat dies jedoch (noch?) nicht geführt.

Einfluss auf die künftige Entwicklung des Informatikunterrichts dürften die einheitlichen Prüfungsanforderungen der KMK in der Abiturprüfung Informatik (EPA Informatik) gewinnen, schließlich beschreiben sie die Anforderungen und damit die Kompetenzen, über die die Abiturienten am Ende ihrer Schulzeit verfügen sollen (vgl. [4]). Die EPA Informatik wurden auf der Grundlage der derzeit geltenden Informatiklehrpläne der 16 Länder erarbeitet. Themen, die den Lern- und Prüfungsbereichen der EPA Informatik nicht zuzuordnen sind, können bis zu einem Drittel einer Prüfungsaufgabe ausmachen. Durch diese Öffnung war es nicht notwendig, alle Themen, die in den Ländern derzeit unterrichtet werden, in die EPA aufzunehmen. Damit war es bei deren Erarbeitung möglich, inhaltliche Schwerpunkte zu setzen. Die Entwicklung der Schulinformatik wird vermutlich auch weiterhin dynamisch verlaufen. Die EPA wurden daher so formuliert, dass es nicht erforderlich ist, sie alle drei Jahre zu überarbeiten. Andernfalls wäre der Wert des Faches Informatik für die Allgemeinbildung kritisch zu hinterfragen. EPA legen auch fest, welchen Schwierigkeits- und Komplexitätsgrad Abituraufgaben haben sollen. Diese Festlegungen haben sowieso längerfristigen Bestand. Die EPA Informatik betonen, der fachdidaktischen Diskussion der letzten Jahre folgend, das Modellieren im Informatikunterricht. So werden als grundlegende Modellierungstechniken im Sinne von Techniken der geistigen Arbeit objektorientierte Modellierung, Datenmodellierung, zustandsorientierte Modellierung, Modellierung von Abläufen mit Algorithmen, funktionale Modellierung und regelbasierte Modellierung konkret aufgeführt.

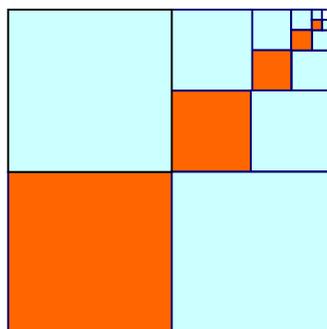
Das Verhältnis von Modellieren und Programmieren betrachtet der Autor als noch nicht ausdiskutiert. Nach seiner Auffassung geht es im Informatikunterricht sowohl um das eine als auch um das andere.

H. Wedekind forderte im Rahmen seines Lehrauftrags „Informatik als Grundbildung“ im Sommersemester 2004 in Jena, Sprache und Logik als Grundlagen des Informatikunterrichts anzusehen². Dieser Ansatz ist es Wert, auf seine schulische Umsetzbarkeit näher untersucht zu werden (vgl. [14]). Als „Minimal-Folgerung“ ist zu formulieren: Mehr Logik in die Schule!

Immer wieder wird nach den für den Einsatz an allgemein bildenden Schulen am besten geeigneten Programmiersprachen gesucht. Ich bezeichne die Suche als das Programmiersprachenproblem und wundere mich – ich gebe gern zu, dass ich hier etwas übertreibe –, dass noch niemand vorgeschlagen hat, die Standards, Systeme und Sprachen HTML, WWW, CSS1, CSS2, DSSSL, XSLT, XSL, CGI, SSI, JSP, ASP, SSJS, PHP, EJB, SSL, DOM Level 1, XML, DTD, XML Schema, XPath, XHTML, XLink, XPointer, Unicode, CORBA, Java, C# und JOSH der Reihe nach im Informatikunterricht abzuhandeln. Kaum ein Lehrer (vermute ich) wäre dazu zwar in der Lage, die Schüler würden daher scharenweise weglaufen, aber *modern* wäre das Vorgehen allemal. Bei den Diskussionen wird häufig vergessen, dass es nicht Aufgabe von allgemein bildenden Schulen ist, direkt und unmittelbar auf das Berufsleben vorzubereiten. Als Konsequenz sind Kriterien aufzustellen, anhand derer eine Einschätzung von Programmiersprachen erfolgen kann. Dazu soll ein Vorschlag unterbreitet werden:

1. Die Programmiersprache eignet sich aufgrund ihrer Struktur für den Schulunterricht, d. h. sie besitzt eine geringe Komplexität oder sie ist in ihrer Komplexität skalierbar einsetzbar.
2. Mit der Programmiersprache lassen sich Probleme aus verschiedenen Gebieten bearbeiten.
3. Vor dem breiten Einsatz einer Programmiersprache wurden Erfahrungen zumindest an einigen Schulen gewonnen.
4. Für die Schülerinnen und Schüler liegen geeignete Unterrichtsmaterialien vor (z. B. Schulbücher, Materialien im WWW).
5. Auf der Grundlage der an den allgemein bildenden Schulen erlernten Programmiersprachen soll ein Einarbeiten in andere Sprachen möglich sein.
6. Die Programmierumgebung sollte kostenlos verfügbar sein.

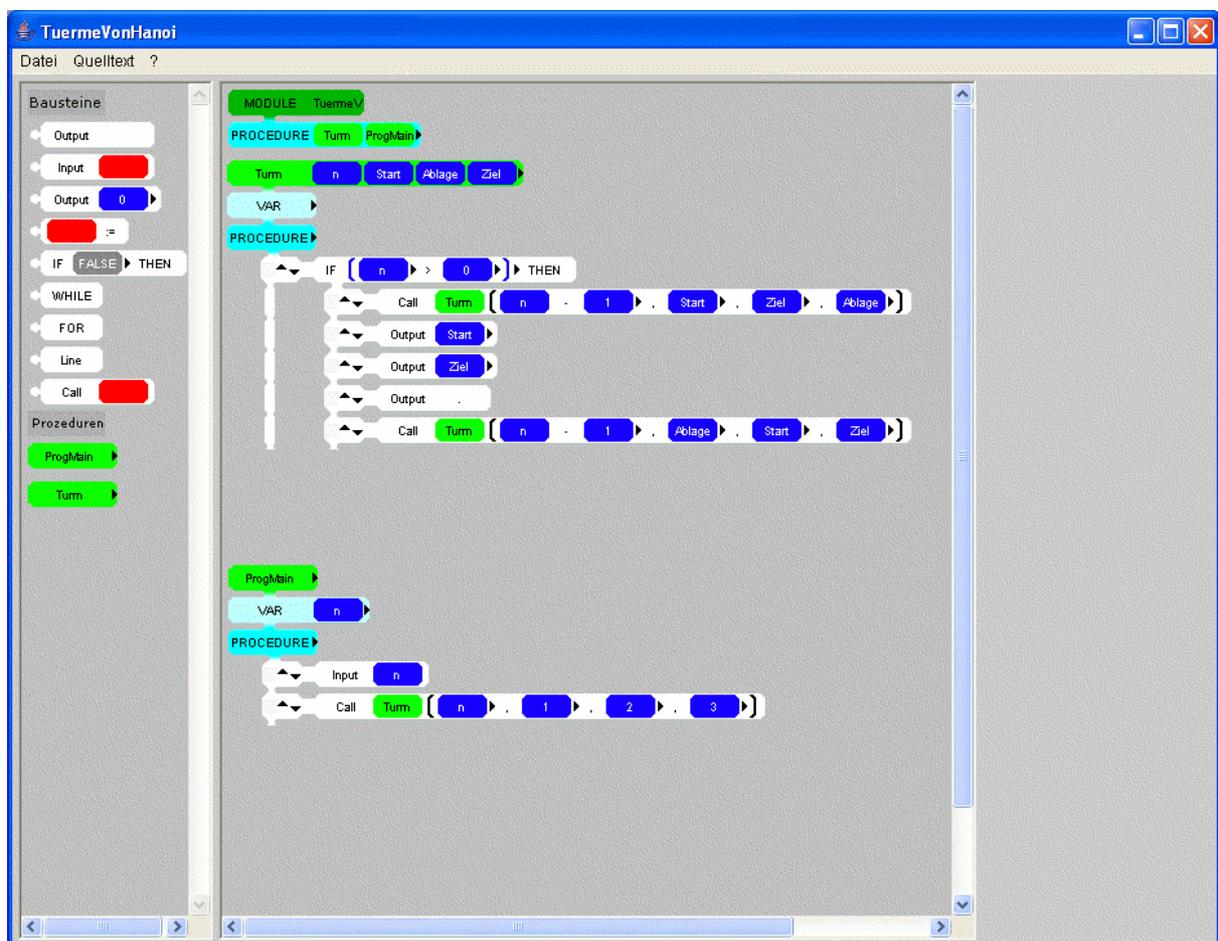
Mit Interesse las der Autor vor 15 Jahren einen Aufsatz des österreichischen Computerpioniers H. Zemanek über das Prinzip LOOK! (vgl. [15]). Ein Beispiel für LOOK! liefert die folgende Abbildung, die in der Topic Study Group 16 (Visualisation in the teaching and learning of mathematics) von ICME-10 präsentiert wurde (4.-11. Juli 2004 in Kopenhagen):



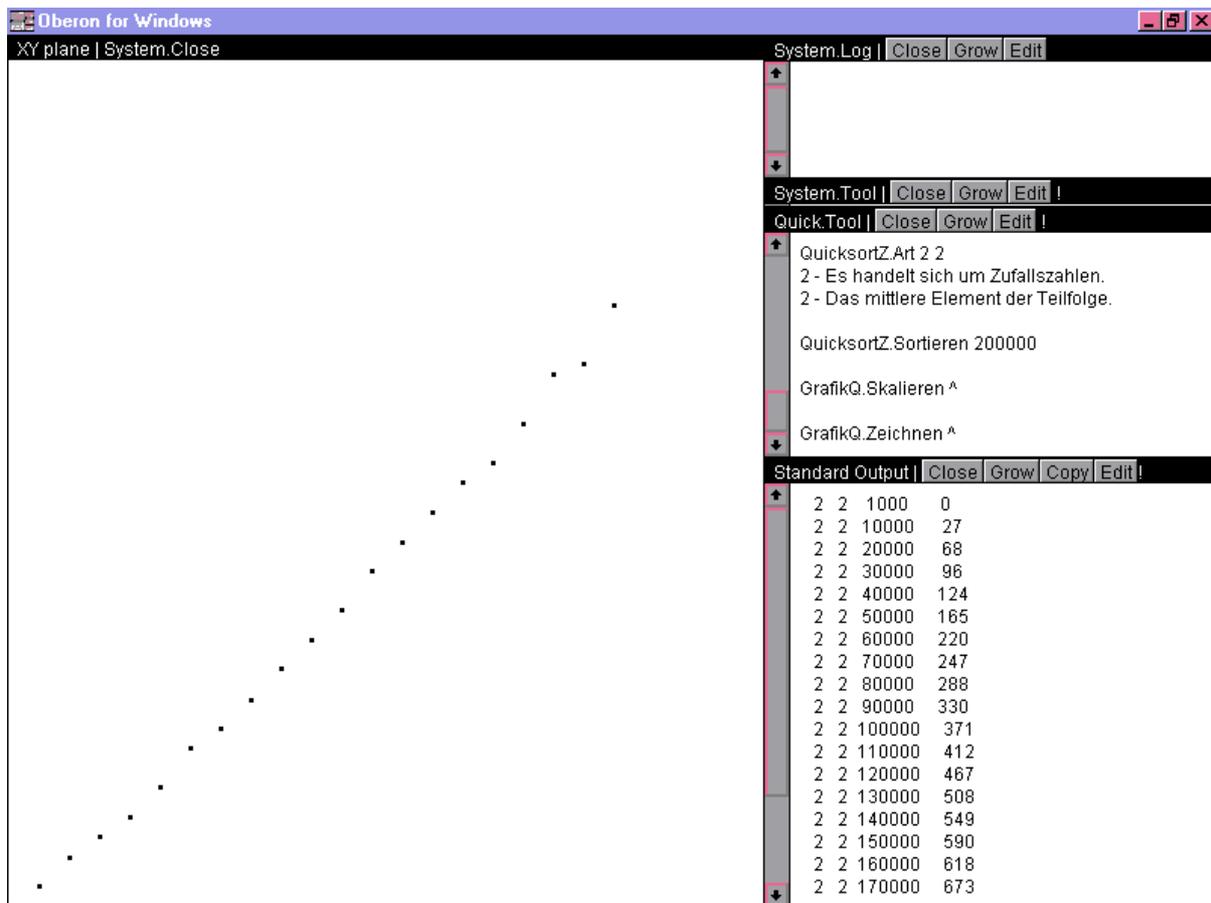
² H. Wedekind wies in seinen Vorlesungen nachdrücklich auf Gottlob Frege (1848-1925) hin, den Begründer der modernen Logik, der jahrzehntlang in Jena wirkte.

An der Abbildung lässt sich der Wert der entsprechenden unendlichen geometrischen Reihe $\frac{1}{4} + (\frac{1}{4})^2 + (\frac{1}{4})^3 + (\frac{1}{4})^4 + (\frac{1}{4})^5 + \dots$ ablesen³.

In dem Zusammenhang kann die Studienarbeit von L. Kohl genannt werden, in der es um visuelle Programmiersprachen für den Einsatz an Schulen geht. In der Arbeit wurden existierende Systeme analysiert, es wurden Informatik-Lehrkräfte zu ihren Erfahrungen und Vorstellungen zu dem Thema interviewt und es wurden Anforderungen an eine visuelle Programmiersprache für den Einsteiger spezifiziert (vgl. [13]). In der darauf aufbauenden Diplomarbeit werden Design und Implementierung des Systems „Puck“ vorgestellt und es wird über erste Erfahrungen bei dessen Einsatz an Schulen berichtet. Ein visuelles Programm wird aus Bausteinen zusammengefügt, die formschlüssig ineinander passen. Zahlreiche Fehlermöglichkeiten sind damit ausgeschlossen. Aus dem visuellen Programm wird der Quelltext automatisch generiert (exemplarisch in Oberon-2). Das folgende mit Puck entwickelte Programm löst das bekannte Problem „Türme von Hanoi“:



³ Die Lösung wird hier nicht verraten.



8. Ausblick

Ein Schwerpunkt der Forschung im Rahmen der Casio-Stiftungsprofessur soll das Lehrerhandeln im Informatik- und Mathematikunterricht sein. An fünf Gymnasien wird das folgende Szenarium zur externen Unterstützung der Informatiklehrerinnen und -lehrer erprobt: Erarbeiten der Kompetenzbeschreibung für ein Thema, Erstellen von Testaufgaben, Übergabe der Testaufgaben an interessierte Schulen, Schreiben und Auswerten des Tests durch die unterrichtenden Lehrkräfte, Erarbeiten von Schlussfolgerungen für den eigenen Unterricht durch die beteiligten Lehrerinnen und Lehrer. Die eigentlichen Testergebnisse sind durch die Schulen nicht weiterzumelden. Vielmehr werden die Lehrerinnen und Lehrer zu Nutzen, Aufwand, Problemen und Varianten des Vorgehens befragt. Konkreter Gegenstand ist das Thema „Rekursion und Iteration“, das Relevanz für die Abiturprüfung im Fach Informatik besitzt (vgl. [4]). Die Untersuchungen stellen einen fachspezifischen Beitrag in der aktuellen Diskussion zu Bildungsstandards dar.

Ein Mathematiklehrer untersucht derzeit die Planungsphase eines Mathematikunterrichts mit CAS genauer. Er ist jeweils zur Hälfte an der Universität und an einem Gymnasium tätig. Dies ist eine Möglichkeit, Schulbezogenheit in der fachdidaktischen Forschung zu gewährleisten. In den Untersuchungen wird von der These ausgegangen, dass der Prozess der Entwicklung der Professionalität der Lehrenden unterstützt werden kann, indem man die Bereitschaft und die Kompetenz zu selbstkritischer Arbeit und zu kommunikativer, kooperativer und öffentlich wirksamer Arbeit befördert (vgl. [10]). Wesentliche Grundlage für die wissenschaftliche Arbeit sind vielfältige eigene Unterrichtserfahrungen.

Und zum Abschluss die schlichte Antwort auf die Frage „Unterricht – bald nur noch mit Computer?“. Ja.

Dank auch an dieser Stelle an die Casio Europe GmbH für das Stiften der Professur an die Friedrich-Schiller-Universität Jena.

Literatur

- 1 Baumann, R.: Zeitaufwand von Sortierverfahren. In: LOG IN Heft Nr. 130 (2004), S. 51-55
- 2 Clocksin, W. F.; Mellish, C. S.: Programmieren in Prolog. Springer Berlin, Heidelberg, New York 1990
- 3 Dietz, P.: Menschengleiche Maschinen. Wahn und Wirklichkeit der künstlichen Intelligenz. Bühler und Heckel Berlin 2003
- 4 Einheitliche Prüfungsanforderungen in der Abiturprüfung Informatik. Beschluss vom 1.12.1989 i. d. F. vom 5.2.2004. Luchterhand München, Neuwied 2004
- 5 Einheitliche Prüfungsanforderungen in der Abiturprüfung Mathematik. Beschluss vom 1.12.1989 i. d. F. vom 24.5.2002. Luchterhand München, Neuwied 2003
- 6 Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen. GI-Fachausschuss „Informatische Bildung in Schulen“. Beilage von: LOG IN 20 (2000) Heft 2
- 7 Fothe, M.: Antwortkontrollalgorithmen für Reaktionsgleichungen der anorganischen Chemie. In: Dresdner Reihe zur Forschung 15/1986, S. 132-137
- 8 Fothe, M.: Rettet, was zu retten ist! ... und das andere, das lasst sein. In: LOG IN 20 (2000), Heft 6, S. 72
- 9 Fothe, M.: Zeitverhalten von Sortierverfahren – Beispiele für experimentelles Arbeiten im Informatikunterricht. In: Hubwieser, P. (Hrsg.): Informatische Fachkonzepte im Unterricht. 10. GI-Fachtagung Informatik und Schule INFOS 2003. Lecture Notes in Informatics, Bonn 2003, S. 111-120
- 10 Fothe, M. (Hrsg.): Mathematikunterricht und Computer – Bestandsaufnahme und Ausblick. Bericht von der Tagung am 24./25. September 2004. In: Jenaer Schriften zur Mathematik und Informatik 11/04 (im Internet verfügbar)
- 11 Frindte, W.; Stauche, H.: Abschlussbericht zur wissenschaftlichen Begleitung der TSCN-Medienzentren (Kurzfassung). FSU Jena 2004 (Institut für Psychologie, Institut für Erziehungswissenschaften)
- 12 Klieme, E. u. a.: Zur Entwicklung nationaler Bildungsstandards. Eine Expertise. BMBF 2003
- 13 Kohl, L.: Konzepte der visuellen Programmierung und ihrer Einsatzmöglichkeiten an Schulen. Studienarbeit an der FSU Jena 2004 (Fakultät für Mathematik und Informatik)
- 14 Wedekind, H.; Ortner, E.; Inhetveen, R.: Informatik als Grundbildung, Teil III: Gleichheit und Abstraktion. Informatik Spektrum 4/2004, S. 337-342
- 15 Zemanek, H.: Der Geist in der Flasche: Warum der Computer nicht ausschaut. In: Informationstechnik – it Heft 1/1988, S. 3-10

Wenn man keinen Durchblick hat,
ist man froh, wenn's weitergeht.
(Prof. Dr. Klaus Brunnstein, Lehrerfortbildung
am 30.11.2001 in Erfurt)