

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

Magnus Rabel, Reinhard Oldenburg

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

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

1 Motivation der Untersuchung

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

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

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

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

2 Untersuchungsmethode

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

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

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

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

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

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

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

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

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

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

3 Ergebnisse und Interpretation

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

Auswertung der Konzeptbewertungen

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

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

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

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

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

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

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

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

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

¹ Vgl. die grau unterlegten Zellen in Tabelle 1

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

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

Durch lineare Strukturgleichungsmodelle [Fo06] können Beziehungen zwischen Variablen erkundet werden. Dazu wurden die Summenvariablen $Algo = b_{Fall} + b_{Schleife} + b_{Proz} + b_{Rek}$, $DS = b_{Array} + b_{Liste} + b_{Stapel}$, $OO = b_{ObjAtt} + b_{ObjKla} + b_{AssAgg} + b_{Vererb}$ jeweils für die Bewertung zum Verständnis und zur Entwicklung (gekennzeichnet durch den angehängten Großbuchstaben) getrennt berechnet. Durch hypothesengeleitetes Variieren ergab sich das in Abb. 1 gezeigte Strukturmodell, das die beste Modellpassung (adjusted Model fit nach R/sem: 0.75) liefert. Man erkennt, dass die Bewertungen von Datenstrukturkonzepten und von algorithmischen Konzepten stark zusammenhängen. Im Vergleich dazu ist die Bewertung der Konzepte der Objektorientierung nur schwach angekoppelt.

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

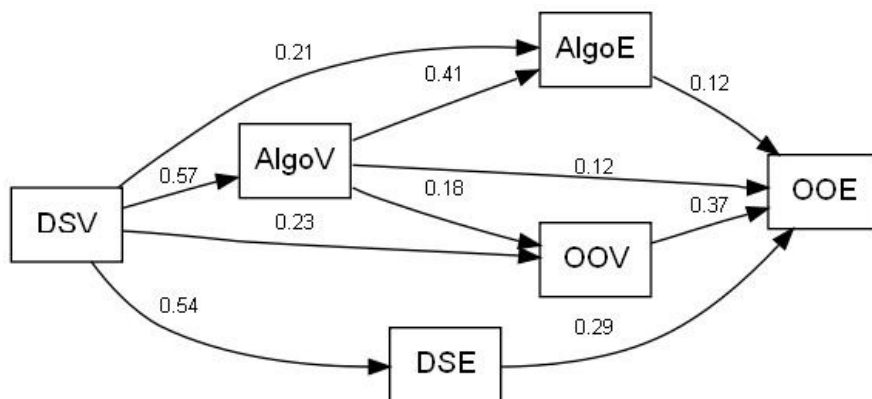


Abbildung 1: Strukturmodell

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

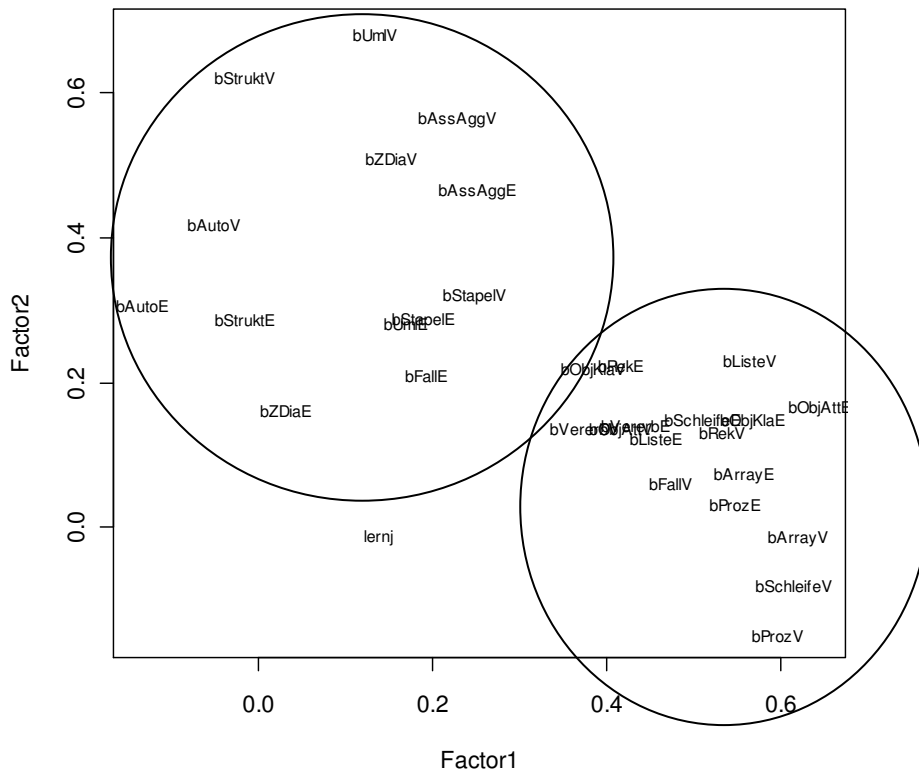


Abbildung 2: Faktorenanalyse

Auswertung der Modellnennungen

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

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

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

Auswertung der durchgeführten hoch bewerteten Projekte

Insgesamt waren die häufigsten Nennungen:

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

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

Auswertung der Wünsche nach Projekten, die noch nicht durchführbar sind

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

Insgesamt waren die häufigsten Nennungen:

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

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

4 Schlussfolgerungen

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

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

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

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

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

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

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

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

Literaturverzeichnis

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

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

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