

# GI-Edition

## Lecture Notes in Informatics

**Andreas Schwill, Carsten Schulte,  
Marco Thomas (Hrsg.)**

## Didaktik der Informatik

**3. Workshop der GI-Fachgruppe  
„Didaktik der Informatik“  
19.–20. Juni 2006  
an der Universität Potsdam**

Gesellschaft für Informatik (GI) publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in co-operation with GI and to publish the annual GI Award dissertation.

Broken down into the fields of

- Seminar,
- Proceedings
- Dissertations
- Thematics

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

The volumes are published in German or English

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

ISSN 1617-5468

ISBN 978-3-88579-193-5

The intention of the DDI Workshop organized by the GI special interest group on Didactics of Informatics is to contribute to the professionalization of Didactics of Informatics and to support research in the field of informatics education.

This volume contains contributions accepted for the third workshop at the University of Potsdam, which mainly focuses on empirical studies in informatics education.







Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.)

## **Didaktik der Informatik**

### **3. Workshop der GI-Fachgruppe „Didaktik der Informatik“**

**19.-20. Juni 2006  
an der Universität Potsdam**

Gesellschaft für Informatik 2006

## **Lecture Notes in Informatics (LNI) - Proceedings**

Series of the Gesellschaft für Informatik (GI)

Volume P-99

ISBN 978-3-88579-193-5

ISSN 1617-5468

### **Volume Editors**

Prof. Dr. Andreas Schwill

Universität Potsdam, Didaktik der Informatik

August-Bebel-Str. 89, D-14482 Potsdam

e-mail: [schwill@cs.uni-potsdam.de](mailto:schwill@cs.uni-potsdam.de)

Jun.-Prof. Dr. Carsten Schulte

Freie Universität Berlin, Didaktik der Informatik

Takustr. 9, D-14195 Berlin

e-mail: [schulte@inf.fu-berlin.de](mailto:schulte@inf.fu-berlin.de)

Prof. Dr. Marco Thomas

Universität Münster, Institut für Didaktik der Mathematik und der Informatik

Fliednerstr. 21, D-48149 Münster

e-mail: [thomasma@uni-muenster.de](mailto:thomasma@uni-muenster.de)

### **Series Editorial Board**

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

Jörg Becker, Universität Münster, Germany

Ulrich Furbach, Universität Koblenz, Germany

Axel Lehmann, Universität der Bundeswehr München, Germany

Peter Liggesmeyer, TU Kaiserslautern und Fraunhofer IESE, Germany

Ernst W. Mayr, Technische Universität München, Germany

Heinrich Müller, Universität Dortmund, Germany

Heinrich Reinermann, Hochschule für Verwaltungswissenschaften Speyer, Germany

Karl-Heinz Rödiger, Universität Bremen, Germany

Sigrid Schubert, Universität Siegen, Germany

### **Dissertations**

Dorothea Wagner, Universität Karlsruhe, Germany

### **Seminars**

Reinhard Wilhelm, Universität des Saarlandes, Germany

© Gesellschaft für Informatik, Bonn 2006

**printed by** Köllen Druck+Verlag GmbH, Bonn

## **Vorwort**

Der 3. Workshop der Fachgruppe „Didaktik der Informatik“ der Gesellschaft der Informatik e.V., der vom 19.-20.06.2006 an der Universität Potsdam stattfand, setzte die erfolgreiche Reihe von Workshops zur Didaktik der Informatik fort, die im Jahre 2002 mit dem Workshop in Bommerholz begann.

Der 3. Workshop sollte zum Erfahrungsaustausch über alle Fragen der Didaktik der Informatik dienen und widmete sich erneut ausgewählten Fragestellungen dieses Themenkomplexes, die durch eingereichte Beiträge intensiv behandelt wurden. Erwünscht waren vor allem aktuelle Forschungsbeiträge zur Didaktik der Informatik, besonders aus dem Bereich empirischer Forschung, an Schulen und Hochschulen. Eingereichte Beiträge wurden nach wiss. Standards begutachtet und sollten

- eine klare theoretische Fundierung aufweisen, die auf existierender Literatur zur Didaktik der Informatik oder von Bezugswissenschaften aufbaut,
- explizit den Beitrag der Arbeit zum fachdidaktischen Wissen über das Lehren und Lernen informatischer Inhalte herausstellen.

Beide Ziele sind mit den akzeptierten und hier veröffentlichten Beiträgen eindrucksvoll erreicht worden.

Weitere Informationen zum Workshop können im Internet unter der Adresse

<http://www.informatikdidaktik.de/FachgruppeDDI2006>

abgerufen werden.

Unser besonderer Dank gilt den hier nicht genannten Helfern für den großartigen Einsatz bei der Vorbereitung und Durchführung des Workshops.

Potsdam, im Oktober 2006

Andreas Schwill, Carsten Schulte, Marco Thomas



## **Inhaltsverzeichnis**

Extreme Programming – Extreme Learning? Erfahrungen aus dem Grenzbereich zwischen Arbeiten und Lernen.....	7
Anette Brüggemann, Matthias Rohs	
Entwicklung einer softwaregestützten Unterrichtsevaluation im Gesamtkontext des Einsatzes von mobilen Endgeräten im Informatikunterricht .....	17
Hendrik Büdding	
Beitrag des Informatikunterrichts zur Entwicklung von Medienkompetenzen.....	29
Stefan Freischlad	
Mobilkommunikation im Informatikunterricht – Thesen und empirische Ergebnisse .....	39
Gerrit Kalkbrenner, Benedikt Schultebrucks, Martin Sawatzki	
Ein Test- und Analyseverfahren zur Kontrastierung von Problemlöse-Prozessen informatischer Hoch- und Niedrigleister – erste Ergebnisse einer Pilotstudie.....	49
Bertold Kujath	
Standards für die Mittelstufeninformatik? .....	71
Eckart Modrow	
Begriffliche Strukturen der Informatik: Ein empirischer Zugang.....	79
Markus Schneider	
Unterrichtsmodellentwicklung zur Förderung des Informatiksystemverständnisses mit Entwurfsmustern .....	89
Peer Stechert	



# Extreme Programming – Extreme Learning? Erfahrungen aus dem Grenzbereich zwischen Arbeiten und Lernen

Anette Brüggemann, Matthias Rohs

Fachbereich Pädagogik  
Helmut-Schmidt-Universität, Universität der Bundeswehr  
Holstenhofweg 85, 22043 Hamburg  
anette.brueggemann@hsu-hh.de  
matthias.rohs@hsu-hh.de

**Abstract:** Extreme Programming (XP) ist ein Ansatz der Softwareentwicklung, der nicht nur ein neues Paradigma für die Programmierung, sondern scheinbar ganz „unbeabsichtigt“ und „nebenbei“ auch noch eine stark lernhaltige Arbeitsform darstellt. Dieser Artikel ist eine erste Annäherung an XP aus der Perspektive des (informellen) Lernens und zeigt anhand praktischer Beispiele das große Potential für die Kompetenzentwicklung, das sich hinter den Prinzipien des Extreme Programming verbirgt.

## 1 Go-IT – aber wie?

Go-IT ist der auffordernde Name eines Modellversuchs zur Entwicklung von Weiterbildungsinstrumenten für die IT-Branche. Der Fokus des Modellversuchs auf kleine und mittlere IT-Unternehmen begründet sich darin, dass hier der Großteil der Beschäftigten arbeitet und zudem die vorrangige wirtschaftliche Expansion der Branche stattfindet. Außerdem hat sich gezeigt, dass sich die Arbeitsprozesse in KMU deutlich von denen groß-industrieller Unternehmen hinsichtlich der Flexibilität, Dynamik und der Ganzheitlichkeit der Aufgaben unterscheiden.

Weiterbildung in der IT-Branche war und ist aber mehr als das Lernen im Rahmen von organisierter Aus- und Weiterbildung, sondern ein Prozess, der vor allem in der Arbeit stattfindet. Go-IT widmet sich aus diesem Grund der Erforschung und Entwicklung von Arbeits- und Lernformen, die formelles und informelles Lernen miteinander verbinden. Informelles Lernen unterscheidet sich dabei gegenüber formellem, organisiertem Lernen dadurch, dass es aus den allgemeinen Arbeitszusammenhängen entsteht, auf konkrete Problemstellungen bezogen ist und nicht pädagogisch unterstützt wird [De05]. Im Modellversuch werden diesbezüglich „intentionale Lernformen“ (Lernen durch Arbeitsaufgaben, Multiplikationsformen) und „lernhaltige Arbeitsformen“ (Tandemformen, Projektteamarbeit) unterschieden (vgl. Abb. 1). Ziel des Modellversuchs ist es, in Zusammenarbeit mit den projektbeteiligten IT-Unternehmen Konzepte zu entwickeln, wie diese Lern- und Arbeitsformen unterstützt werden können.

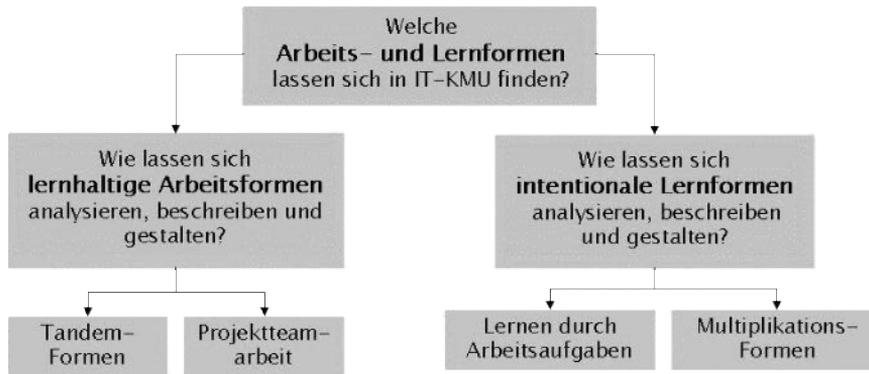


Abbildung 1: Fragestellungen im Modellversuch Go-IT

## 2 Extreme Programming – eine Revolution?

Bei der Suche nach lernhaltigen Arbeitsformen sind wir auf „Extreme Programming“ (XP) gestoßen. XP ist eine Methode der Softwareentwicklung, deren auffälligstes Merkmal ist, dass stets zwei Entwickler/innen gemeinsam an einer Aufgabe und einem Rechner programmieren<sup>1</sup>. Ein Umstand, der zumindest auf dem ersten Blick betriebswirtschaftliche Bedenken auslöst. Aber auch darüber hinaus scheint mit dieser Art der Softwareentwicklung ein Paradigmenwechsel verbunden. Doch was steckt dahinter?

Ausgangspunkt ist der große Umfang an gescheiterten Softwareentwicklungsprojekten. Zwischen 40% und 50% wird die Quote der Projekte geschätzt, die nicht die gesetzten Ziele erreichen. Bei etwa 80 % der Entwicklungsprojekte wird die veranschlagte Zeit deutlich überschritten [Si04]. Die Ursache dafür dürfte auch im klassischen Entwicklungsprozess von Analyse, Design, Implementierung und Test liegen, der wenig Dynamik aufweist und Kunden nur unzureichend in den Entwicklungsprozess einbezieht. Durch XP soll dies verbessert und gleichzeitig eine höhere Qualität in kürzerer Zeit als üblich erreicht werden.

Erste wissenschaftliche Erkenntnisse sowohl zur betriebswirtschaftlichen Effizienz dieser Arbeitsform als auch zur Lernförderlichkeit und Kompetenzentwicklung finden sich bei der amerikanischen Forscherin LAURIE WILLIAMS von der North Carolina State University. Sowohl zeitliche, wie auch qualitative und motivationale Aspekte scheinen nach ihren Untersuchungen für die Arbeit in Tandems/Paaren zu sprechen: „In their first assignment (I call the "jelling-assignment") the pairs spent 60% more programmer hours than the individuals. In the second assignment, they had gotten used to this pair-

---

<sup>1</sup> Extreme Programming beinhaltet neben Pairprogramming eine ganze Reihe weiterer Praktiken, wie Test-Driven-Development, Customer Test, Collective Code Ownership und andere, auf die wir in diesem Artikel nicht näher eingehen werden. Eine kurze Übersicht dazu bietet [www.xprogramming.com](http://www.xprogramming.com)

programming thing. The pairs spent only 20% more total time than the individuals. By the third assignment, the pairs spent only 10% more time – so if an individual spent 10 hours on the assignment, the pair worked together for 5 hours and 15 minutes. In all cases, the pairs passed about 15% more of the post-development test cases. And, over 90% say they enjoy programming more and they feel more confident in their work when pairing.” (<http://www.extremeprogramming.org/stories/pair6.html> , 30.1.2006)

Kent Beck [Be03], Begründer des „Extreme Programming“, hat vier Werte für die Softwareentwicklung definiert, die den Rahmen von XP bilden: Einfachheit, Feedback, Kommunikation und Mut. Die „Einfachheit“ bezieht sich auf die angestrebten Lösungen und den Entwicklungsprozess, das „Feedback“ auf Tests sowie die schnelle Rückkopplung mit dem Kunden und den Anwender/innen. Die „Kommunikation“ steht für den intensiven Austausch im Team. „Mut“ ist notwendig, sich auf die Umsetzung dieser Werte einzulassen [LRW02, S. 4f]. Diese Werte werden durch 15 Prinzipien näher beschrieben [ebd. S. 5ff].

### 3 Extreme Programming – Extreme Learning?

Was hat XP nun mit Lernen zu tun? Wenn man den Anwender/innen Glauben schenken mag, reichlich wenig. Zwar wurde XP nicht explizit zu Lernzwecken entwickelt, dennoch steckt in dieser Softwareentwicklungsmethode ein hohes Lernpotential, in der das informelle Lernen noch recht unreflektiert einen breiten Platz einnimmt. Sie wird dadurch zu einer interessanten Arbeitsform zur gezielten Unterstützung von Lernprozessen. Um die Lernhaltigkeit von XP deutlich zu machen, erscheint uns zunächst der Vergleich der XP-Prinzipien mit denen einer „lernförderlichen Arbeitsgestaltung“ hilfreich.

Schon seit geraumer Zeit beschäftigt sich insbesondere die Arbeits- und Organisationspsychologie mit der Erforschung von Dimensionen, die die Lernförderlichkeit der Arbeit als auch der Arbeitsumgebung beschreiben [vgl. u. a. Be96; FK87; So96]. Mit der Zielsetzung einer umfassenden beruflichen Handlungskompetenz sind dabei unter anderem folgende Kriterien zu nennen:

- **Ganzheitliche Handlung:** Die Arbeitsaufgaben bilden den Arbeitsprozess von der Vorbereitung bis zur Kontrolle ab (Vollständige Handlung) und werden unter ökonomischen, ökologischen und sozialen Gesichtspunkten betrachtet.
- **Handlungsspielraum:** Die Freiheitsgrade bei der Ausführung der Arbeitsaufgabe, sowie die Partizipations- und Mitgestaltungsmöglichkeiten sind hoch.
- **Problem- und Komplexitätserfahrung:** Die Arbeitsaufgabe zeichnet sich u. a. durch Vielschichtigkeit, Unbestimmtheit, Vielfalt und verschiedene Zielsetzungen aus.
- **Soziale Unterstützung:** Kolleg/inn/en unterstützen sich gegenseitig, wobei dieser Prozess durch Teamarbeit zusätzlich gefördert wird. Die Möglichkeit zur Kommunikation ist eine wesentliche Voraussetzung dafür.
- **Individuelle Entwicklung:** Die Herausforderungen der Arbeit orientieren sich am Entwicklungsstand der Mitarbeitenden und führen weder zu Über- noch zu Unterforderung.

- **Feedback:** Die Mitarbeitenden, Vorgesetzten und Kunden geben sich regelmäßig Rückmeldungen über die Qualität der Arbeit.

Betrachtet man nun die Werte und Prinzipien des XP, so zeigen sich deutliche Überschneidungen zu den Kriterien der Lernförderlichkeit. An einigen Beispielen soll dies nun verdeutlicht werden.

Schon drei der vier zentralen Werte des Extreme Programming sprechen für die hohe Lernhaltigkeit des Ansatzes:

**Kommunikation:** Nach den Grundsätzen des XP sind alle Projektmitglieder aufgefordert intensiv (!) miteinander zu kommunizieren. Dabei wird das persönliche Gespräch als effektivster Weg des Austauschs hervorgehoben. Pair Programming als grundlegende Arbeitsform und der regelmäßige Wechsel der Paare fördern den intensiven Austausch zwischen den Mitarbeitenden, so dass sogar auf die sonst übliche prozessbegleitende Dokumentation verzichtet werden kann.

**Feedback:** Über Rückmeldungen wird beim Extreme Programming die Qualität der Softwareentwicklung sichergestellt. Durch eine enge Zusammenarbeit mit dem Kunden, der nach Möglichkeit bei der Programmierung dabei sein sollte, sowie durch schnelle Testfolgen von Systemversionen beim Anwender wird ein zeitnahes und prozessbegleitendes Feedback ermöglicht. Kurzfristige Rückmeldungen zu den Arbeitsergebnissen werden dabei auch vor dem Hintergrund angestrebt, „dass der zeitliche Abstand zwischen Aktion und Feedback eine entscheidende Rolle für den Lernerfolg spielt.“ [LRW02, S. 6].

**Mut:** Der Mut, als dritter Wert, wird als Grundlage für die anderen Werte betrachtet. „Einfache Lösungen erfordern Mut, weil sich herausstellen könnte, dass die Lösung zu einfach war. (...) Feedback erfordert Mut, weil eine negative Bewertung als persönliche Kritik aufgefasst werden könnte. (...) Kommunikation erfordert Mut, weil sich herausstellen kann, dass man einem Missverständnis aufgesessen ist und etwas an der erstellten Software wieder ändern muss.“ [LRW02, S.5]. Was hier sehr konkret gefasst wird, erfordert eine Unternehmenskultur, die den Mut auch zulässt, die tolerant mit Fehlern umgeht, Kompetenzlücken akzeptiert und damit das für die Offenheit notwendige Vertrauen schafft. Dieses Vertrauen erfordert wiederum Mut von der Geschäftsführung, sich auf diese Prozesse auch einzulassen. Mut ist damit auch als Synonym für eine lernförderliche Unternehmenskultur zu verstehen, die die Grundlage für das Handeln nach den XP-Werten und Prinzipien bildet.

**Einfachheit:** Mit diesem Wert wird u. a. das Bestreben nach einfachen Lösungen adressiert. Dies ist zunächst keine Voraussetzung für die Kompetenzentwicklung, Jedoch wird damit ein Anspruch erhoben, der unter Umständen eine intensivere Auseinandersetzung mit den Anforderungen aus technischer Sicht als auch aus Kundensicht erfordert. Somit wird das Lernen darüber noch mal angeregt.

Einige XP-Prinzipien sollen an dieser Stelle noch angesprochen werden, da sie nicht nur die Umsetzung der Werte konkretisieren, sondern auch das intentionale Lernen in dieser Arbeitsmethode belegen.

- **Lernen lehren:** Dieses Prinzip bezieht sich auf die Zielsetzung, die Programmierern zur selbständigen Umsetzung von XP zu befähigen und zeigt damit die stärkste Lernintention aller Prinzipien.
- **Verantwortung übernehmen:** Hinter diesem Prinzip steht die Überzeugung, dass Verantwortung Mitarbeitenden nicht zugewiesen, sondern von ihnen übernommen (eingefordert) wird. Die Übernahme von Verantwortung bedeutet gleichzeitig die Ausweitung des Handlungsspielraums und der Erfahrungsmöglichkeiten und unterstützt damit das Lernen.
- **Gezielte Experimente:** Experimente dienen dazu, schnell Klarheit darüber zu erlangen, ob eine Entscheidung richtig oder falsch war. Damit verhindern sie nicht nur „Fehlentwicklungen“ im Projekt, sondern tragen auch gezielt zum Erkenntnisgewinn bei.
- **Offene, aufrichtige Kommunikation:** Dieses Prinzip stellt gerade aus der Perspektive des Lernens eine wesentliche Voraussetzung dar und adressiert gleichzeitig ein zentrales Problem des Wissensmanagement. Ohne die Offenheit, Wissen weiterzugeben wie auch Kompetenzlücken einzuräumen, ist gemeinsames Lernen kaum möglich.

Insgesamt zeigt sich so, dass Lernen im XP durch viele Werte und Prinzipien unterstützt und gefördert wird und hohe Überschneidungen mit den Kriterien einer lernförderlichen Arbeitsgestaltung aufweist. Auf der Grundlage dieser Betrachtungen ist Extreme Programming als eine stark lernförderliche Methode der Softwareprogrammierung einzuschätzen. Doch wie ist es in der Umsetzung? Zeigt sich die Lernförderlichkeit auch in der Praxis?

Um dies herauszufinden, schien es uns notwendig, diese Arbeitsform in der Praxis zu untersuchen. Überraschenderweise stellte es sich als schwierig heraus, ein Unternehmen zu finden, das nach den Prinzipien des Extreme Programming arbeitet. Wie uns auch durch Experten bestätigt wurde, schlägt sich die Faszination für XP nur selten in einer vollständigen Umsetzung der Werte und Prinzipien im Arbeitsalltag nieder. Dennoch wird in vielen Firmen mit einzelnen Prinzipien gearbeitet, wie etwa mit dem zeitweisen Programmieren in Tandems oder mit schnellen Release-Zyklen.

## **4 XP vor Ort – und das funktioniert wirklich? Gesammelte Eindrücke und Gedanken**

Schließlich haben wir eine Softwareentwicklungs-Firma gefunden, die uns Gelegenheit zur wissenschaftlichen Untersuchung gegeben hat. Dazu wählten wir den Ansatz des „Shadowing“, der Beobachtung mit anderen qualitativen Methoden verbindet [Mc05], um uns dem Untersuchungsgegenstand zu nähern. Konkret wurde eine zweistündige Beobachtung von zwei „Pairs“, wie sich die Arbeitstandems nennen, durchgeführt und die daraus gewonnenen Erkenntnisse in einem anschließenden Gespräch mit den Programmierern verifiziert. Anregungen bildeten hierbei die oben genannten wissenschaftlichen Erkenntnisse von L. Williams zum Pairprogramming.

Die Firma arbeitet seit Beginn ihres Bestehens nach den XP-Prinzipien. Für den Arbeits-

alltag bedeutet das konkret, dass vor Ort beim Kunden in kleineren Teilprojektteams zwei bis vier Pairs, miteinander an einem Teilprojekt arbeiten. Innerhalb dieser Teilprojekte wechselt die Zusammensetzung der Paare häufig, spätestens aber nach einer Woche.

XP bedeutet aber auch, sich innerhalb fast rigide anmutender Rahmenbedingungen zu bewegen: die Arbeit muss zur gleichen Zeit begonnen und beendet werden. Gleiches gilt für die Pausen. Um die enge Kundenbindung sicherzustellen, arbeiten die Mitarbeiter/innen für die Dauer des Projekts direkt beim Kunden.

Beeindruckt waren wir über die intensive und hochgradig kommunikative Zusammenarbeit der Pairs, die sich aus der Arbeitsform ergibt. Ständiges lautes Denken, Fragen stellen und Ideen entwickeln („Ach stimmt“, „Genau!“, „Mein Problem ist...“, „nee, da hast du was vergessen“, „Machst Du das immer so?“, „Ach so“, „Mir ist wirklich nicht klar...“, „lass uns mal das probieren...“) lassen selten Pausen von mehr als einer halben Minute entstehen. Ohne diese ständige Verbalisierung der eigenen Gedanken wäre es den Partnern nicht möglich, das Vorgehen des anderen zu verfolgen und zu verbessern. Daher ist es auch notwendig, die eigenen Gedanken zumindest so weit innerlich zu ordnen, dass sie explizierbar sind, wobei dieser Prozess eher unbewusst abläuft. Individuelles ausprobieren erfolgt daher auch an gesonderten Arbeitsplätzen.



Abbildung 2: XP in der Anwendung

Es scheint eine ganz besondere sozial-kommunikative Kompetenz gefragt zu sein, das eigene Handeln permanent zu kommentieren, zu erklären. Weitere Beispiele dieser Kommunikations- und Arbeitsform lassen sich auch in anderen Berufsfeldern finden. In einem Flugzeug-Cockpit etwa wird immer zu zweit der Check vor dem Abflug gemacht und gleichzeitig kommentiert. Auch bei Chirurgenteams im Operationssaal werden die nächsten Handlungen jeweils laut verbalisiert.

Hintergrund dieses Vorgehens scheinen die gleichen Ziele zu sein, die mit dieser kommunikativen Arbeitsform verfolgt werden:

- **Besseres Qualitätsmanagement:** gegenseitige Kontrolle, Fehler erkennen und korrigieren
- **Höhere Aufmerksamkeit:** durch den Zwang zum Verbalisieren die innere Aufmerksamkeit und Konzentration erhöhen
- **Komplementarität:** sich ergänzen in den Fähigkeiten: wenn die eine Person nicht weiter weiß, kann die andere helfen.

Anders als in den beschriebenen Beispielen besteht beim XP aber die bereits angedeutete Regel oder auch Kultur des „Muts“, der Kreativität und Fehlerfreundlichkeit im Sinne einer Ermunterung, auch unübliche Wege zu gehen und auszuprobieren. Das Kommentieren dient also nicht nur dem Kontrollieren sondern auch der Entwicklung von Innovationen. Ebenso ist die Methode auch für Einarbeitungsphasen einsetzbar, so dass hier Kompetenzunterschiede kein Hinderungsgrund sind. Deutlich wurde aber von den Expert/innen darauf hingewiesen, dass die Kompetenzniveaus eher ähnlich sein müssen, um wirklich effektiv und befriedigend arbeiten zu können.

## 5 Kompetenzentwicklung und XP – eine neue Orientierung?

Legt man den Kompetenzbegriff mit der Unterteilung in fachliche, soziale und personale Kompetenz mit dem Ziel einer umfassenden beruflichen Handlungskompetenz zugrunde, so lassen sich für die Betrachtung von XP als Arbeits- und Lernform einige Einschätzungen ableiten.

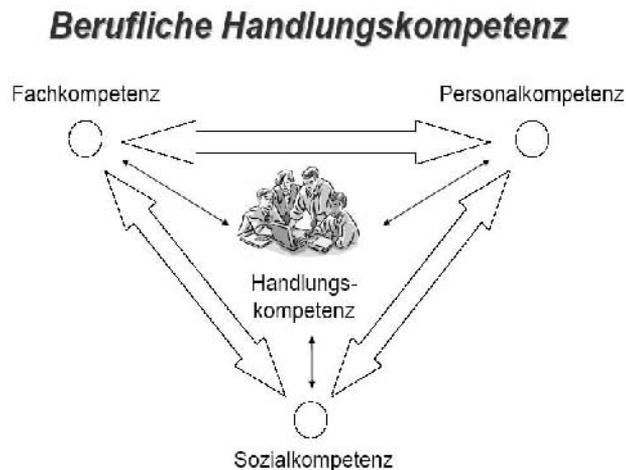


Abbildung 3: Berufliche Handlungskompetenz

Es ist davon auszugehen, dass die **fachliche Kompetenz** durch die Art der Zusammen-

arbeit gefördert wird. Fachliches Know-how wird permanent ausgetauscht und somit erweitert. Ich lerne die Vorgehensweise des anderen kennen und profitiere von seinem auch informellen Erfahrungswissen. Der Austausch bleibt dabei aber auf den konkreten Arbeitsgegenstand bezogen. Nach Aussage der Mitarbeiter/innen ist das Arbeiten mit XP eher mit einem soliden handwerklichen Arbeiten verbunden, als mit dem Ziel des technisch Machbaren.

Sehr wahrscheinlich entwickeln sich auch **soziale Kompetenzen** im Laufe der Zeit sowohl durch ein „Modell-Lernen“ weiter, als auch durch die permanente Übung und den häufigen Wechsel der Partner/innen. Darüber hinaus wurde von der Geschäftsleitung auch darauf hingewiesen, dass eine sorgfältige Personalauswahl für das Arbeiten mit Extreme Programming gerade im Hinblick auf die sozialen und personalen Kompetenzen von besonderer Bedeutung ist.

Die **personale Kompetenz** wird u.a. durch die Anforderung im XP gefördert, mich permanent selbst in Frage zu stellen, bzw. mit großer Offenheit einem anderen den Stand meiner Erfahrungen und kommunikativen Fähigkeiten preiszugeben. Das bestätigen die Mitarbeitenden deutlich: man gibt sich durch diese Arbeitsform eine unmittelbare fachliche und personale „Blöße“. Nur eine Kultur der Wertschätzung, Fehlerfreundlichkeit und Kommunikation weiß diese zu „schützen“ und mit ihr umzugehen. Gleichzeitig erfordert die gemeinsame Arbeit an einem Produkt ein hohes Maß an Kompromissbereitschaft, sowie eine hohe Frustrations- und Ambiguitätstoleranz.

Die sozialen und personalen Kompetenzen stellen somit eine wesentliche Voraussetzung für die Arbeit mit XP dar. Der sorgfältigen Personalauswahl kommt aus diesem Grund eine große Bedeutung zu. Gleichzeitig bietet Extreme Programming aber auch die Möglichkeit zur Entwicklung eines „organisationalen“ Kompetenzniveaus, d. h. dass es durch die Arbeit in Paaren insbesondere zu einer Nivellierung fachlicher Kompetenzen kommt und ein hoher allgemeiner Wissenstand über projekt-/unternehmensbezogene Prozesse erreicht wird. Andeutungsweise lässt sich dies auch für sozialen und personalen Kompetenzen feststellen, die sich durch diese Art der Zusammenarbeit angleichen.

## 6 Ausblick und offene Fragen

Die Auseinandersetzung mit arbeitsintegrierten Formen des Lernens in der IT-Branche steht erst am Anfang. Wie sich am Beispiel des Extreme Programming zeigt, haben sich aus den Anforderungen der Arbeit und den lange Zeit weitgehend fehlenden Antworten auf den Weiterbildungsbedarf in dieser hoch innovativen Branche eigene Lösungen entwickelt. Die Auseinandersetzung mit diesen Arbeitsformen verspricht nicht nur ein besseres Verständnis des Lernens in der IT-Branche, sondern ermöglicht es auch, Kompetenzentwicklung in diesem Feld besser zu unterstützen.

Nach den bisherigen Ergebnissen unserer Arbeit stellt Extreme Programming ein spannendes und lohnendes Feld für die Lern- und Weiterbildungsforschung und -beratung dar. Lernen wird auf verschiedene Weise durch eine lernförderliche Arbeitsgestaltung unterstützt.

Dieser Artikel stellt jedoch nur eine erste Annäherung an XP als lernhaltige Arbeitsform dar. Mit den gewonnenen Erkenntnissen haben sich neue Fragen für die berufliche Weiterbildung und die Personal- und Organisationsentwicklung in IT-KMU ergeben:

- **Zusammenspiel zwischen Strukturierung und Selbststeuerung:** Die Zusammenarbeit der Pairs erfordert auf der einen Seite eine sehr rigide Strukturierung der Arbeitsabläufe. Auf der anderen Seite arbeiten die Pairs weitgehend selbst gesteuert. Ein wenig erinnerte das Setting an Open Space Konferenzen, in denen genau die Koppelung von Struktur und Selbststeuerung als Garant für eine aktive Zusammenarbeit auf Tagungen gesehen wird. Welche Strukturen braucht die Selbststeuerung um erfolgreich zu sein?
- **Planung der Arbeit:** Im Gegensatz zum „klassischen“ Vorgehen bei der vollständigen Handlung, in der der sorgfältigen Information und Vorbereitung von Arbeitsprozessen ein besonderer Stellenwert zukommt, ist das Arbeiten der Pairs beim Programmieren weitgehend durch den Moment und kurze iterative Planungszyklen bestimmt. Individuelle Vorgehensstrategien müssen während der Programmierung gedanklich durchdrungen und zur Nachvollziehbarkeit des Vorgehens und zur Diskussion dem anderen Pair verbalisiert werden. Längerfristige Planungen der Arbeitsprozesse beruhen weitgehend auf Selbstbeobachtung (z.B. Ermittlung der "velocity" des Teams). Werden durch dieses Vorgehen wichtige Orientierungspunkte für das Arbeiten genommen? Welche Auswirkungen hat diese Form der Planung auf die Arbeit und das Lernen?
- **Gestaltung der Arbeitsumgebung:** Durch die Programmierung vor Ort beim Kunden und die wechselnden Paarzusammenstellungen haben die Programmierer/innen keinen festen Arbeitsplatz. Dieser Übergangscharakter wurde in der kaum vorhandenen (individuellen) Gestaltung der Arbeitsumgebung deutlich. Die spartanische Einrichtung hätte es ermöglicht, den Arbeitsplatz innerhalb von Sekunden spurlos zu verlassen. Steht diese Gestaltung der Arbeitsumgebung nicht im Widerspruch zu einer angenehmen, lernförderlichen Arbeitsatmosphäre? Oder unterstützt es eine kreative „Laboratmosphäre“?

Im Rahmen des Modellversuchs Go-IT soll diesen Fragen in der weiteren Arbeit nachgegangen werden. Interessant erscheint hier auch ein breiterer Blick auf die Tandem-Arbeitsform in der IT-Branche, etwa die auffällig häufig bestehenden Geschäftsführungstandems, ihre Arbeits- und Lernformen und die Kriterien, die sie – formell und informell – für eine erfolgreiche Zusammenarbeit entwickelt haben. Durch Interviews, die konkrete Prozessbegleitung in der Praxis und weitere Shadowing-Phasen sollen die Erkenntnisse vertieft werden. Die Ergebnisse dazu werden auf der Website des Projekts veröffentlicht ([www.go-it.net](http://www.go-it.net)).

## Literaturverzeichnis

- [Be03] Beck, K.: Extreme Programming – Das Manifest. Addison-Wesley. München, 2003.  
[Be96] Bergmann, B.: Lernen im Prozeß der Arbeit. In (Arbeitsgemeinschaft Qualifikations-Entwicklungs-Management Berlin Hrsg.): Kompetenzentwicklung '96: Strukturwandel und Trends in der betrieblichen Weiterbildung. Münster, 1996; S. 153-262.

- [De05] Dehnbostel, P.: Informelles Lernen in betrieblichen und arbeitsbezogenen Zusammenhängen. In (Künzel, K. Hrsg.): Internationales Jahrbuch der Erwachsenenbildung: Informelles Lernen – Selbstbildung und soziale Praxis, Böhlau, Köln, 2005; S. 143-164.
- [FK87] Franke, G.; Kleinschmitt, M.: Der Lernort Arbeitsplatz. Berlin, Köln, 1987.
- [LRW02] Lippert, M.; Rook, S.; Wolf, H.: Software entwickeln mit XP eXtreme Programming-Erfahrungen aus der Praxis. dpunkt-Verlag, Heidelberg, 2002.
- [Mc05] McDonald, S.: Studying actions in context: a qualitative shadowing method for organizational research, Qualitative Research, Vol. 5, No. 4, 455-473.
- [Si04] Sixtus, Mario: Gemeinsam auf die Spitze treiben, in; Die Zeit 01/2004, Online im Internet: <http://www.zeit.de/2004/01/T-Extremprogrammierer> (15.01.2006)
- [So96] Sonntag, Kh.: Lernen im Unternehmen. Effiziente Organisation durch Lernkultur. München. 1996.

# Entwicklung einer softwaregestützten Unterrichtsevaluation im Gesamtkontext des Einsatzes von mobilen Endgeräten im Informatikunterricht

Hendrik Büdding

Arbeitsbereich Didaktik der Informatik  
Institut für Didaktik der Mathematik und Informatik  
Fliednerstr. 21  
48147 Münster  
budding@uni-muenster.de

**Abstract:** Seit PISA hat die Qualitätsforschung des schulischen Unterrichts im deutschen Bildungswesen einen besonders hohen Stellenwert. In der Lehramtsausbildung an der Universität Münster wird vertieft auf die Schwerpunkte Selbstevaluation und Reflektion im Lehrerberuf eingegangen. Dies betrifft auch Informatik-Lehramtsstudierende und -Lehrer, die erlernen, wie man Unterrichts-Evaluationen plant und durchführt, um die selbst gewählten, didaktisch und inhaltlich angestrebten Ziele effizient zu evaluieren und eine Qualitätsentwicklung im eigenen Informatik-Unterricht zu erzielen. In der Praxis brauchen Lehrer ein einfaches Werkzeug, um ihren Unterricht entweder selbst oder innerhalb der Fachschaft evaluieren zu können. Am Beispiel einer Anfangs-Unterrichtssequenz zum Thema „Computer Hard- und Software“ mit Integration von Handheld-Computern im Informatikunterricht werden Erfahrungen einer Unterrichtsevaluation unter Verwendung der kostenlosen Umfragesoftware GrafStat in Kombination mit (Handheld-) Computern als eine mögliche Alternative zu gebräuchlichen Print-Feedbacks vorgestellt.

## 1 Einleitung

Spätestens seit Pisa besteht die Notwendigkeit, den eigenen Unterricht kritisch zu hinterfragen und ihn konstruktiv zu evaluieren. Dabei sind global durchgeführte Evaluationen für den einzelnen Lehrer nicht so aufschlussreich, als wenn er seinen eigenen Unterricht mit seinen gruppenspezifischen Methoden und eigenen Lernzielen evaluiert und anschließend optimiert. Doch wie und womit wird eine Unterrichtsevaluation entwickelt? Wie bettet man eine Evaluation in den Unterricht ein? Welche sekundären, inhaltlichen Möglichkeiten bietet die Thematik einer Evaluation oder Umfrage?

Anhand einer beispielhaften Informatik-Unterrichtseinheit mit Realschülern in der Erprobungsstufe (Klasse 5/6), welche die Integration von mobilen Endgeräten (Handheld-Computer / PDA= Personal Digital Assistant) beinhaltet, wird im Verlauf dieses Artikels eine entsprechende Feedback-Realisierung dargestellt. Die ersten Ergebnisse des Einsatzes der verschiedenen Inhalte und Methoden werden unter den Aspekten der Forderung

von Bildungsstandards, wie sie im Heft der Computerzeitschrift „Log In“ (Heft 135 (Thema „Standards in der informatischen Bildung“)) und auf der Tagung „Informatik und Schule“ (INFOS) 2005 gefordert wurden, beschrieben.

Es wird verdeutlicht, dass Handheld-Computer ebenso wie elektronische Feedbacks problemlos in den Informatik-Einführungsunterricht integriert werden können.

## **2 Softwaregestützte Qualitätsentwicklung im Informatikunterricht**

Im Rahmen der Schul- und Unterrichtsevaluationen finden an deutschen Schulen immer mehr interne und externe Untersuchungen statt, um ein faktisches Feedback zu erhalten. Nicht erst durch PISA und TIMSS beschäftigen sich immer mehr Lehrer mit der Frage, wie der Einzelne – unabhängig von außen – sich und seinen Unterricht selbst bzw. besser evaluieren kann.

Bislang befindet sich die deutsche Informatikdidaktik-Forschung in der Standardisierungsforschung noch in der Phase, sich auf Mindeststandards zu verständigen, welche bundesländerübergreifend gelten sollen. Dazu regt H. Puhlmann (et al.) [Pu05] die Formulierung von Bildungs- bzw. Mindeststandards für das Fach Informatik an. Da bislang keine informatischen Bildungsstandards aus den Kern-Curricula, wie z.B. in Deutsch, Mathematik und der ersten Fremdsprache, abgeleitet wurden, steht in der Informatik ein konsensbildender Prozess noch aus.

Somit ist bisher jede Lehrkraft innerhalb ihrer Fachschaft unter Berücksichtigung der allgemeinen Bildungsstandards, der Forderungen des Fach-Curriculums und der Richtlinien selbst dazu aufgefordert, entsprechende Bildungsziele zu definieren und sie mit entsprechenden Items (Fragen oder Aufgaben) abzufragen. Zusammen mit den Schulnoten etc. ist es dann möglich, den aktuellen Stand der Schüler in seinen Facetten zu erfahren und ggf. entsprechende Maßnahmen zur Intervention zu treffen. Um den eigenen Standpunkt als Informatiklehrerin bzw. Informatiklehrer in der informatischen Bildungslandschaft genau zu lokalisieren, bedarf es Standards. Diese Orientierungshilfen ermöglichen es auch, sich durch Vergleichbarkeit zu messen und somit zu orientieren und die Inhalte einander zu nivellieren.

Durch die individuelle Wahl der Items kann eine individuelle Rückkopplung erzeugt werden, die mit entsprechenden Unterrichtsgesprächen und Leistungstest nicht erfasst werden kann. Dabei besteht die Möglichkeit, dass der Lehrkörper entweder autonom seinen Informatikunterricht oder zusammen mit den Kollegen der Fachschaft den gesamten Informatikunterricht an der Schule systematisch evaluiert.

Durch vermehrt auftretende, regelmäßige Evaluationen in den schulischen und pädagogischen Bereichen, sind fundierte Aussagen z.B. über Lernprozesse, spezifische Lernschwierigkeiten, Fachleistungen, Diagnosemethoden und Hochbegabung möglich [KR00]. Dabei sind Längsschnitt- und Querschnittsdiagnosen der Schüler im Unterricht für die Lehrerin bzw. den Lehrer, aber auch für die Forschung der Informatikdidaktik, von hohem Wert [BD95]. Dabei lassen sich Aspekte interdisziplinärer Ergebnisse für die

Schulinformatik transformieren und ermöglichen es, die Forschung auf dem Gebiet der Informatikdidaktik und der Entwicklung von Bildungsstandards voranzutreiben. Dies sollte wieder an die Informatiklehrerinnen und -lehrer als Schulungen, Materialien und Erfahrungen zurückfließen, um selbstständig Unterrichtsevaluationen schnell und effizient durchzuführen.

Um eine Qualitätsverbesserung zu erreichen, werden zunächst Möglichkeiten aufgezeigt, eine Längs- und Querschnittanalyse effizient durchzuführen und Möglichkeiten eines Feedbacks effektiv zu nutzen. Es werden im weiteren Verlauf die Instrumente vorgestellt, die Lehrer und Schüler für Umfragen, Feedbacks und Evaluationen verwendeten.

## **2.1 Nutzung der Umfragesoftware GrafStat zur Selbstevaluation**

Um eine schnelle, aber dennoch wirkungsvolle Reflektion des Unterrichtes durchzuführen, bietet sich die Verwendung einer Software an, welche die routinemäßig anfallenden Aufgaben effizient unterstützt, um auch eine zeitnahe Auswertung der erhobenen Daten zu gewährleisten. Der Einsatz der Softwarelösung GrafStat (Grafik und Statistik) bietet der Lehrerin oder dem Lehrer die Möglichkeit der strukturierten Unterstützung bei der Durchführung einer eigenen Unterrichtsbefragung/Evaluation.

Von Vorteil ist die Softwarelösung als Hilfswerkzeug für Unterrichtsbefragungen. GrafStat wurde speziell für den Schuleinsatz entwickelt, was man im leicht zugänglichen und übersichtlich strukturierten Aufbau positiv spürt. Die Oberfläche von GrafStat ist dabei aufgebaut wie der Ablauf einer Umfrage bzw. Unterrichtsevaluation: Fragebogen planen und anlegen, ihn gestalten, die Daten erfassen, die Daten auswerten, analysiert präsentieren und die Befragungen verwalten. Die gut strukturierte Software ermöglicht jedem Benutzer – auch ohne weitergehende Statistik-Kenntnisse – auf effiziente Art und Weise, diese Phasen und ihre Ideen zu realisieren. GrafStat ist durch langjährige Entwicklung im schulischen Umfeld zu einem interdisziplinären und leistungsfähigen Programm geworden. Neben der Realisierung von Schulprojekten im Bereich der empirischen Sozialforschung, ist die Software multifunktional gestaltet, so dass auch semi-professionelle Unterrichtsevaluationen damit durchgeführt werden können.

Der Mehrwert von GrafStat liegt dabei in der vielfältig einsetzbaren Software im edukativen Bereich. In der Vergangenheit lag der Einsatzschwerpunkt der Software im Bereich der projektbezogenen, selbstständigen Schülerarbeit als empirischer Sozialforscher [We05]. Neben der Unterstützung zur Unterrichtsevaluation und um Unterrichtsqualität zu optimieren, kann GrafStat zur Erstellung von Multiple-Choice-Tests (Abfrage von Anwenderwissen), zur Erstellung von Fragebogenprojekten im Bereich der Softwareentwicklung oder Usability-Analyse verwendet werden.

Die im weiteren Verlauf angesprochene beispielhafte Unterrichtsevaluation im Fach Informatik wurde mit dem Umfragewerkzeug GrafStat und der multimedialen Lehr- und Lernumgebung “Forschen-mit-GrafStat“ realisiert. Auf der Doppel-CD “Forschen-mit-GrafStat“ befinden sich neben der Software GrafStat kleine Tools, Anleitungen und viele Projektausarbeitungen mit erfolgreichen interdisziplinären Umfragebögen. Die Foren der Internetseite des CD-ROM-Projektes [www.forschen-mit-grafstat.de](http://www.forschen-mit-grafstat.de) wurde ebenfalls für

das Selbststudium herangezogen.

## **2.2 Nutzung des Handheld-Computers im Informatikunterricht zur Unterrichts-evaluation**

Neben der im folgenden Kapitel angesprochenen Integration der Handheld-Computer-Technologie als Lerninhalt und Lernwerkzeug in die Schulinformatik, ist die Verwendung der internetfähigen WLAN-Handheld-Computer als Umfragewerkzeug für Unterrichtsfeedbacks ebenfalls interessant.

Die Verwendung von Onlinebefragungen bietet sich im Computerraum oder mit einem internetfähigen PDA in einem Klassenraum an. Es vermeidet auch die Angst der Schüler, der Lehrer könnte ihre Handschrift erkennen; somit ist die Befragung deutlich anonymer als Print-Fragebögen. In einer Printbefragung oder in einem persönlichen Feedback mit einem PDA könnte z. B. der Klassensprecher als unabhängige Instanz fungieren und die Befragung koordinieren.

Eine Verbesserung des Unterrichtsfeedbacks stellt aus der Erfahrung des Autors die effiziente und effektive Online-Umfrage dar. Zwar bekommt der Lehrer auch ein Feedback über Gespräche, Klausuren, Tests und andere Noten, aber ein unabhängiges, zeitnahes Feedback des Schülers ist nur über eine anonyme Online-Befragung möglich. Die bisherigen Erfahrungen des Autors mit Unterrichts-Evaluationen haben gezeigt, dass bei feedback-geübten Schülerinnen und Schülern in ca. 5 Minuten etwa 15-20 Fragen (incl. Freiantworten) beantwortet werden können.

## **3 Die Informatik-Unterrichtssequenz, in der die Unterrichts-Evaluation eingebettet ist**

In einem Anfangsunterricht des Schulfachs Informatik wurden die Schülerinnen und Schüler an die Benutzung der Hard- und Software von Desktop-Systemen und Handheld-Computern herangeführt.

Mit einer konkreten Hardwarepräsentation wurden die verschiedenen Elemente, die beim Eingabe-Verarbeitung-Ausgabe-Prinzip (E-V-A-Prinzip) verwendet werden, vorgestellt. Die Schülerinnen und Schüler reflektierten das Gesehene und Gehörte in der Klasse, beispielsweise lernten die Schülerinnen und Schüler mit Hilfe eines Übungsblattes mit Zuordnungsaufgaben anhand von Symbolen zu abstrahieren. Ob Funkmaus oder miniaturisierte Tastatur, ob 5- oder 3,5-Zoll-Laufwerk, die Schüler verbanden schnell die fotorealistischen Abbildungen (mit Untertiteln) mit den entsprechenden, abstrahierten Hardware-Klassifikations-Bildern.

In einem Hardware-Zuordnungsspiel mit den Schülerinnen und Schülern, bei der „gezeigte“ Hardware den Feldern „Eingabe“, „Verarbeitung“ und „Ausgabe“ zugeordnet wurde, kamen einige „Objekte“ vor, die die Schüler selbst aus Prospekten herausgeschnitten hatten, aber nicht eindeutig einer „Klasse“ zuzuordnen waren. Auf einer

Wandtapete waren dafür „Klassen“-Flächen aufgemalt worden, in der die Schüler ihre „Objekte“ einordnen und mit einem Klebestift aufkleben sollten. Doch sie hatten erkannt, dass sich in dem Handheld-Gerät (portable Spielekonsole und ein Handheld-Computer) ein Computer verbarg, aber die Zuordnung in eine „Klasse“ nicht eindeutig möglich war.

Anschließend wurde der praktische Umgang mit den Computern, das Starten des Rechners und das Einloggen mit der Schulkennung über das MS-Windows-Betriebssystem in das Netzwerk geübt. Sie lernten die Bedienung der Tastatur und der Maus sowie die Verwendung von Disketten und Druckern. Sie entdeckten die Grundzüge des Grafikprogramms MS-Paint und bearbeiteten digitale Fotos. Dabei sollten sie die Fotos so bearbeiten, dass sie auf der Grundlage ihres eigenen Fotos sich selbst darüber abmalen und dabei die (tendenziell) gleichen Farben vom Original für Haare, Augen, Mund, etc. verwenden. In einem Unterrichtsspiel wurden dann die selbst gemalten Bilder den Personen zugeordnet. Die grundlegenden Merkmale vom „Original“-Menschen und dem abstrahierten, selbst gemalten „Objekt“-Bild, hatten die Schüler dabei schnell entdeckt.

In der evaluierten Unterrichtssequenz traf sich die Gruppe zu jedem Stundenanfang, um das Erlernte der vergangenen Stunde zu reflektieren. So entstand mit jeder Stunde eine ruhigere Arbeitsatmosphäre und der Einstieg in den Unterricht gelang zunehmend schneller. Am Ende der Phase wurden der Inhalt der aktuellen Stunde und die anstehenden Aufgaben vorgestellt sowie eventuelle Verständnisprobleme in der Aufgabenstellung direkt geklärt.

### **3.1 Einsatz des Handheld-Computers als Lernwerkzeug und Lerninhalt**

Im Rahmen des Anfangsunterrichts wurden die Schülerinnen und Schüler an die Benutzung der Hard- und Software herangeführt. Dabei wurde innerhalb dieser Unterrichtseinheit neben der Einführung in einen Desktop-Computer auch erste Einführungsversuche in die Welt des Pervasive und Ubiquitous Computing anhand von PDAs erprobt. Die Schule im allgemeinen soll und der Informatikunterricht im speziellen muss Medienkompetenz im Bereich Mobiler Technologien vermitteln, da diese bereits fester Bestandteil der Lebenswelt der Schüler sind. Diese Entwicklung, die bereits in den angelsächsischen Ländern aufgegriffen wurde, muss als wichtiger Bestandteil in den Schulalltag einfließen und sollte für den Informatikunterricht sinnvoll nutzbar gemacht werden [KT05].

Die Miniaturisierung der mobilen Endgeräte und die Evolution der mobilen Technik schreiten voran; diese Entwicklung muss vom Informatik-Unterricht inhaltlich berücksichtigt werden. Daneben sollten die Nutzarmachung dieser „mobilen“ Werkzeuge und die damit verbundenen neuen Inhalte aufgegriffen und ggf. in den Vordergrund gestellt werden. Anhand zwei mitgebrachter PDAs mit Farbdisplay konnten die Schülerinnen und Schüler sehen, probieren und verstehen, was dieses in 2D abgebildete Gerät in Wirklichkeit war und wie groß und schwer bzw. klein und leicht dieser Computer war. Dabei wurde die Integration der Hardware nicht forciert, sondern verlief parallel zur Einführung der Desktop-Geräte.

Trotz des verschiedenartigen Aufbaus der Geräte verstanden die Schüler schnell die Analogien, unterhielten sich und analysierten auch erste Vorteile der Handhelds. Da sie die umfangreichen technischen Möglichkeiten von Computern noch nicht kannten, beschränkte sich ihr Wissen auf tragbare Spielekonsolen, tragbare Musikabspielgeräte und Navigationsgeräte als elektronischen Stadtplan.

### **3.2 Handheld-Computer: Das Fenster in die Welt der Objekte**

Ebenso sollten die Schülerinnen und Schüler eine erste Begegnung mit Fachterminologien erhalten und an die Begriffe wie Objekte, Methoden und Attribute herangeführt werden.

Als Beispiel soll hier das Arbeiten mit dem Grafikprogramm auf Desktop-Computern und PDAs dienen. Immer wieder wurden die Geräte in den Unterrichtsablauf bzw. –verlauf integriert und Analogien der verschiedenen Hard- und Softwaresysteme genutzt. Zwei besonders eifrige Schüler bekamen in der Unterrichtseinheit, in der die Gruppe mit dem MS-Paint Programm auf dem Desktop-PC arbeitete, die PDAs. Sie sollten sich – entsprechend ihrer selbst gemalten Selbstbildnisse auf dem Desktop-PC – auf dem Handheld mit einem PDA-Paint-Programm malen. Durch die Analogie der beiden Softwareprodukte und der intuitiven Stiftbedienung ließ der Lehrer die Schüler die (vereinfachten, also abstrahierten) Objekte (selbst gemalte Selbstbildnisse der Fotos) auf das Display der beiden PDAs malen.

Die einzelnen Unterrichtseinheiten starteten immer am Desktop-PC. Die Schülerinnen und Schüler erlernten am Beispiel der Grafikverarbeitung die Hardwarebedienung und die Hand-Augen-Koordination mit der Maus sowie deren (Inter-)Aktionen auf dem Bildschirm. Bei der Demonstration, mit der Tastatur zu malen, erkannten sie schnell die Grenzen der Tastatur sowie die Vorteile der Maus. In der darauf folgenden Diskussions- und Feedbackrunde wurde demonstriert, dass es auch andere Eingabemöglichkeiten neben der Maus gibt. Der Lehrer und die mit den PDAs arbeitenden Schüler führten nun das PDA-Paint-Programms und dessen Bedienung mit dem Stift vor, der zeigte, welche weitere Eingabemöglichkeit zum herkömmlichen Desktop-PC ein PDA besitzt.

Die beiden selbst gemalten Bilder der Schüler wurden vorgestellt und vom Lehrer kurz die Möglichkeit des Datenaustausches (genannt „Beamen“) über die Infrarot Schnittstelle (Bsp.: Fernsehfernbedienung) demonstriert. Die Bilder wurden weiter bearbeitet und ein Tafelbild zu den Fotos bzw. abstrahierten Selbstbildnissen als Ergebnissicherung erstellt. Durch das Beamen der Bilder änderten sich die „Objekte“ auf den Bildschirmen der PDAs. Man konnte das „Objekt“ des Menschen verändern, indem man ihn mit seinen eigenen Personendaten (z. B. Farben als Attributwerte) wieder neu übermalte und es wieder zurückschickte.

So lernten die Schüler beim Umgang mit den PDAs, aufbauend auf den Desktop-PC-Erfahrungen, spielerisch den Umgang mit den mobilen Endgeräten. Ebenso wurden sie schrittweise an das Fachvokabular des objektorientierten Programmierens herangeführt.

## **4 Aufbau und Durchführung einer Feedbackphase zur Evaluation des Informatikunterrichtes**

Die Lehrperson wollte erfassen, in wieweit sich alte oder neue Techniken ebenso wie Inhalte und Methoden auf die Qualität des Unterrichts auswirken. Da bislang informatische Bildungsstandards nicht klar definiert sind, entwickelte der Lehrende entsprechend den Vorgaben aus der Literatur im Bereich der empirischen Sozialforschung [Bu02, If05], den fachlichen Rahmenbedingungen und allgemeiner Unterrichtsevaluation einen Pilot-Fragebogen, der die spezifischen Erfahrungen der informatischen Unterrichtsreihen für Lehrer untermauern soll.

Anhand von allgemeinen Informationen zum Thema Evaluation, die auf der CD „Forschen mit GrafStat“ abgelegt sind [Sa04], sowie der „Einführung in die Test- und Fragebogenkonstruktion“ von Bühner [Bu04] wurden erste Items (Fragen / Aufgaben) und Itemantworten, die systematische Zusammenhänge wie Korrelation oder Kontingenz aufweisen, entwickelt. Ausgangspunkt für die Item Entwicklung waren u. a. die zehn Merkmale guten Unterrichts von H. Meyer [Me06]. Durch sie lassen sich interessante Aspekte für eigene Informatikunterricht-Feedbackbogen ableiten und in entsprechende Items transformieren. Je nach Interessensschwerpunkt kann die Methodik und der Inhalt in einer eigenen Evaluation schnell durch die festgelegten Fragen eruiert werden.

Aufgrund der ersten Erfahrungen in der universitären Ausbildung versuchte der Lehrer, die eigene Unterrichtsevaluation unter Zuhilfenahme der Umfragesoftware GrafStat zu realisieren. Dabei sollte der gesamte Evaluationsprozess zeitlich sehr kurz gehalten werden, aber dennoch ein entsprechendes Feedback nach den selbst aufgestellten Evaluationszielen liefern [Bu02].

Dazu wurde in einer Feedbackrunde im Schulhalbjahr ein erster Vorab-Fragebogen mit 2 Skalenfragen (ähnlich der Schulnotenvergabe) und einer Frage mit Freiantwort eingeführt, näher erklärt und kurz in Stillarbeit durchgeführt.

Aufbauend auf den ersten Erfahrungen mit dem ausgedruckten Fragebogen, führte der Lehrer eine weitere Vorab-Onlinebefragung via Desktop-PCs durch, worauf eine Online-Feedbackbefragung mit WLAN-fähigen Handheld-Computern folgte.

Das Hauptinteresse lag letztlich aber auf dem Online-Feedback-Fragebogen (Datenaustausch über Browserverbindung), welcher mit den Handhelds direkt in der Feedbackrunde von den Schülern selbst durchgeführt wurde. Dabei wurden die PDAs via WLAN („adhoc Verbindung“) mit einem am LAN hängenden Laptop vernetzt. Die auf den beiden PDAs geöffneten Browser stellten eine mit GrafStat generierte Internetseite als Startseite dar, über welche die Daten auf den beiden PDAs erfasst und auf dem Laptop gesammelt wurden. Bei den Onlineversionen ist eine sehr zeitnahe Auswertung der Daten mit GrafStat möglich, da diese nicht erneut erfasst werden müssen. Somit ist eine erste Auswertung direkt in der Feedbackrunde möglich. Die Feedbackrunde verlängerte sich auf 12 Minuten, gab jedem Schüler aber individuell an, wo im Leistungsspektrum der Klasse er sich befand und wie Mitschüler sich selbst, die Unterrichtseinheiten etc. nach dem Schulnotenprinzip bewerteten. Dies stellte der Lehrer dar, nahm entsprechend

Stellung zu den einzelnen Punkten und diskutierte Veränderungsvorschläge für den zukünftigen Unterricht.

Die Unterrichtsevaluation, die eine nachhaltige Qualitätsentwicklung im Informatikunterricht nach sich ziehen soll, ist in die vier Phasen Vorbereitung, Durchführung, Auswertung (der erhobenen quantitativen und qualitativen Daten) und Präsentation gegliedert. Zu diesen Phasen befinden sich jeweils detaillierte Informationen, aber auch allgemeine, praxiserprobte Beispiele aus interdisziplinären Unterrichtsprojekten auf der GrafStat-Doppel CD.

Grundlage für den Einsatz der web-gestützten Onlinebefragung war u. a. auch, dass die Befragung zur Lerneinheit immer wiederholt werden konnte. Ebenso könnten die Ergebnisse mit vorherigen Durchläufen direkt verglichen werden. Somit ist neben einer Querschnitt- auch eine Längsschnittanalyse einer informatischen Unterrichtseinheit möglich.

Für die fachspezifische Unterrichtsevaluation ist die Aufstellung einer eigenen Indikatorsammlung in den verschiedenen Untersuchungsdimensionen (Bereichen) Ausgangspunkt, welche u. a. auch methodische, (fach-) didaktische und atmosphärische Gesichtspunkte des Unterrichtsgeschehens widerspiegeln soll [Bu99].

Ziel war es, am Ende des Schulhalbjahres den Anfangsunterricht mit der Einführung in die verschiedensten Hard- und Softwareklassen sowie deren Bedienung und das allgemeine Anwenderwissen evaluieren. Neben den kurzen Feedbacksitzungen am Anfang einer Stunde wollte der Lehrer am Schulhalbjahresende eine Unterrichtsevaluation durchführen. Dabei sollte die Akzeptanz und der Mehrwert der verwendeten Methoden und Inhalte analysiert und die Schüler an eine zu entwickelnde Feedbackkultur herangeführt werden.

Die Schüler hatten gelernt, ein Feedback in einer Runde mündlich zu geben, waren im Ausfüllen von Feedbackfragebögen jedoch noch unerfahren. Der Vorteil für ein Feedback über papierbasierte Fragebögen oder über eine Onlinebefragung liegt in der Anonymität der Schülerin bzw. des Schülers. Er ist bei dieser Form des Feedbacks in der Lage, Probleme, Lernhemmnisse oder Fragen zu formulieren, ohne sich vor den Mitschülern outen zu müssen. So lassen sich Methoden-, Inhaltsvariationen und andere Modifikationen innerhalb einzelner Unterrichtseinheiten in ihren Auswirkungen deutlich besser erkennen, als in einem mündlichen Feedback.

Ein weiterer Vorteil ist, dass der Schüler erkennt, dass das Feedback nicht wie eine Klassenarbeit auf ihn zurückfällt, sondern er im Vergleich der Allgemeinheit nicht persönlich auf seine Leistung angesprochen wird (bzw. nicht angesprochen werden kann), er persönlich kennt jedoch seine Antwort und kann sich bei Ergebnisvergleichen selbst einordnen (vergleichbar mit den Publikumsfragen bei „Wer wird Millionär?“). Dabei sollte der Lehrer die Lerngruppe langsam an die Feedback- und Evaluationskultur heranführen und ihr die Wichtigkeit durch entsprechendes Offenlegen der Ergebnisse zeigen.

## 5 Erfahrungen aus dem Qualitätsentwicklungsprozess

Um Innovationsprozesse als Maßnahmen aufgrund eines bestimmten Feedbacks in einer Situation zu erzeugen, sollte man den Qualitätskreislauf nach H.-G. Rolff [Ro01] beim Evaluieren durchlaufen. Der Lehrkörper kann sich orientieren, in welcher Phase des Qualitäts- bzw. Evaluationskreislaufes er sich befindet und wie der nächsten Phase zu begegnen ist. Somit ist es möglich, eine Qualitätsentwicklung im Informatikunterricht anzustreben. Dabei ist es von Vorteil, auf bewährte Materialien aus der Literatur zurückgreifen zu können. Wie Bühner [Bu04] in seinem Buch beschreibt, sollten die Hauptgütekriterien, wie Objektivität, Reliabilität und Validität (S. 28-34) sowie Nebengütekriterien wie Normierung (Standards), Vergleichbarkeit, Ökonomie und Nützlichkeit dann stärker in den Vordergrund gerückt werden.

Die Verwendung eines erfolgreichen Grundframeworks einer Befragung sollte dann entsprechend der eigenen Bedürfnisse modifiziert werden. Werden im Frageblock gleiche Items gewählt, ist es zudem möglich, einen Vergleich der Daten vorzunehmen. Dies ist jedoch aufgrund von divergenten Schul- und Unterrichtssituationen nur bedingt zu empfehlen, da verschiedene Unterrichtsevaluationen meist nur schwer miteinander vergleichbar sind.

Die Online-Befragung mit Desktops oder Mobilien Endgeräten wie Handheld Computern, ermöglicht es dem Befragten, die Daten online auszufüllen und die Auswertung um die Zeit der Dateneingabe zu verkürzen. Dies kann internetbasiert mittels Desktop PCs, Laptops oder PDAs durchgeführt werden. Es ist ebenso möglich, die aktuellen Daten mit bereits erfassten zu koppeln und somit einen entsprechenden Vergleich mit anderen Untersuchungsgruppen zu ermöglichen bzw. der Gruppe die zeitliche Entwicklung direkt aufzuzeigen. Mit GrafStat ist es möglich, eine Online-Befragung aufgrund der vorher eingegebenen Items automatisch generieren zu lassen. Kurzfragebögen können am Ende einer Unterrichtseinheit ausgeteilt werden, Online-Befragungen im Browserfenster bearbeitet werden. Nach einer Feedbackphase kann sich der Lehrer ein umfassendes Bild der Lerner-Situation aller Schüler machen. Dabei bekommt der routinierte, reflektierende Lehrer schnell ein gewünschtes, ggf. anonymes Feedback über Wissensstand, Verständnisschwierigkeiten und Verbesserungsvorschläge. Bei der Online-Befragung entfällt zwar die erneute Dateneingabe in den Computer, dafür ist man bei der Papier-Version nicht an vordefinierte Textfelder gebunden, kann frei schreiben und gegebenenfalls Sachverhalte bildlich darstellen.

Neben den Vorteilen, die GrafStat bei der Unterrichtsevaluation bietet, hat die Verwendung von PDAs den Vorteil, dass mit ihnen zu jeder Zeit, überall und direkt gearbeitet werden kann. Nach dem Einschalten ist der PDA sofort betriebsbereit und die in den Browser geladene Evaluationsseite kann ausgefüllt und z.B. über das WLAN abgeschickt werden. Je nach Unterrichtsevaluation und Computereinsatz ist es somit möglich, individuell den Bogen auszufüllen. Ebenso ist es möglich, innerhalb einer Feedbackrunde parallel die PDAs kreisen zu lassen und ggf. direkt die Daten am Lehrer-Bildschirm auszuwerten, um ein direktes Feedback zu geben und der Meinung einer Datenverfälschung vorzubeugen.

Die Praxiserfahrungen durch das oben erwähnte Beispiel zeigen, dass die softwareunterstützte Durchführung von Feedbacks eine sinnvolle Möglichkeit darstellt. Bei wiederholenden Befragungen (wiederkehrende Unterrichtseinheiten in einer Klassenstufe) ist ein routiniertes Arbeiten möglich und die Option des zeitlichen Vergleichs mit anderen Jahrgängen interessant.

## 6 Zusammenfassung und Aussichten

Wie die genannten Teilaspekte des Beispiels zeigen, ist die Integration von Handheld-Technologien in den Informatikunterricht als Evaluationstool, Lernwerkzeug aber auch als Lerninhalt möglich. Der Einsatz von PDAs muss verstärkt untersucht werden, um die Chancen und Risiken der erweiterten Möglichkeiten unter Berücksichtigung der technologischen und dadurch auch gesellschaftlichen Veränderungen genauer zu eruieren.

Die stärkere Integration von internetfähigen WLAN-Handheld-Computern bedarf großer Forschungsaktivitäten im deutschsprachigen Raum. Dabei sollten nicht die adhoc-Integration von internetfähigen WLAN-Handheld-Computern im schulischen Bereich forciert werden, sondern langfristig geplant und die möglichen Realisierungschancen anhand von ausländischen Studien geprüft werden. Nur durch die Entwicklung von entsprechenden Lernszenarien mit mobilen Endgeräten und aktueller, zielgruppenspezifischer Lernsoftware für den Informatikunterricht im besonderen wird eine wissenschaftlich fundierte Integration dauerhaft möglich und sinnvoll.

Ebenso müssen die Forschungsbestrebungen auf dem Gebiet der Informatikdidaktik in Deutschland verstärkt werden. Nur durch die Entwicklung von Mindest- bzw. Bildungsstandards ist das Schulfach Informatik in der Lage, seine etablierte Stellung im Fächerkanon der Schule weiter auszubauen. Dazu bedarf es der Portierung und Anwendung der interdisziplinären Qualitätsentwicklungswerkzeuge zur verstärkten Optimierung der Unterrichtsqualität im Schulfach Informatik.

## Literaturverzeichnis

- [Bu04] Bühner, M.: Einführung in die Test- und Fragebogenkonstruktion. Pearson Studium Verlag, München, 2004.
- [Bu02] Buhren, C.G. et. al.: Wege und Methoden der Selbstevaluation. Ein praktischer Leitfaden für Schulen. IFS -Verlag, 5.Auflage, Dortmund, 2002.
- [Bu99] Buhren, C.G. et. al.: Qualitätsindikatoren für Schule und Unterricht – Ein Arbeitsbuch. IFS -Verlag, Dortmund, 1999.
- [BD95] Bortz, J.,Döring, N.: Forschungsmethoden und Evaluation für Sozialwissenschaftler. Springer, Berlin, Heidelberg, New York, 1995.
- [If05] Institut für Schulentwicklung Hrsg.: IFS-Schulbarometer. Ein mehrperspektivisches Instrument zur Erfassung der Schulwirklichkeit. IFS -Verlag, 9. überarbeitete Auflage, Dortmund, 2005.
- [KR00] Kempfert, G./ Rolff, H.-G.: Pädagogische Qualitätsentwicklung. Beltz Verlag, Freiburg, 2000.
- [KT05] Kukulska-Hume, A., Traxler, J. Hrsg.: Mobile Learning, A handbook for educators and

- trainers. Routledge, Taylor & Francis Group, London, 2005.
- [Me04] Meyer, H.: Was ist guter Unterricht. Cornelsen Verlag Scriptor, Berlin, 2004
- [Ro01] Rolff, H.-G.: Schulentwicklung konkret: Steuergruppe, Bestandsaufnahme, Evaluation. Kallmeyer Verlag, Seelze-Velber, 2001.
- [Sa04] Sander, W. Hrsg.: Forschen mit GrafStat – Erweiterte Neuauflage 2004. Bundeszentrale für politische Bildung, Bonn, 2004.
- [Pu05] Puhmann, H.: Bildungsstandards Informatik – zwischen Vision und Leistungstests. In (Friedrich, S.): Unterrichtskonzepte für informatische Bildung. Köllen Druck + Verlag, Bonn, 2005; Seite 79 bis 90
- [We05] Weyel, D.: Jugendliche als Sozialwissenschaftler – GrafStat zur statistischen Analyse von Umfragen. In: LOG IN, 136/137. LOG IN -Verlag, Berlin, 2005; S. 124-126



# Beitrag des Informatikunterrichts zur Entwicklung von Medienkompetenzen

Stefan Freischlad

Didaktik der Informatik und E-Learning  
Universität Siegen  
Hölderlinstraße 3  
57068 Siegen  
freischlad@die.informatik.uni-siegen.de

**Abstract:** In diesem Paper wird ein Forschungsprojekt vorgestellt, in dem es um die Anforderungen geht, die der zunehmende Einsatz digitaler Medien im Alltag an den mündigen Bürger stellt und wie diesen Anforderungen im Informatikunterricht begegnet werden kann. Als Rahmenkonzept für ein Unterrichtsmodell wird das Didaktische System „Internetworking“ entwickelt. Das Vorgehen zur Entwicklung des Unterrichtsmodells, des Didaktischen Systems und der Test-Items zur Beschreibung von Kompetenzen wird beispielhaft dargestellt.

## 1 Motivation

### 1.1 Forschungskontext

Seit Jahren besteht die Forderung, dass auch der Informatikunterricht einen Beitrag zur Medienbildung leisten soll (z. B. [GI99]). Es fehlt jedoch ein umfassendes Konzept, das sowohl Ergebnis wissenschaftlicher Forschung ist als auch die Möglichkeit zur direkten Umsetzung im Unterricht bietet.

Das Kulturwissenschaftliche Forschungskolleg „Medienumbrüche“, das durch die Deutsche Forschungsgemeinschaft (DFG) gefördert wird, untersucht die Voraussetzungen und Strukturen des analogen Medienumbruchs zu Beginn des 20. Jahrhunderts und des digitalen Medienumbruchs im Übergang zum 21. Jahrhundert. Das Teilprojekt „Informatikunterricht und E-Learning zur aktiven Mitwirkung am digitalen Medienumbruch“ setzt sich mit der Thematik auseinander, wie Informatikunterricht in der Sekundarstufe und E-Learning in der Erwachsenenbildung einen Beitrag zum Erwerb von Medienkompetenzen leisten können [SSF05].

Im Rahmen des Konzepts zur Medienbildung für die Sekundarstufe II sollen Lernziele und -gegenstände mit den Themenschwerpunkten

A. Strukturen des Internet,

- B. Kommunikationsbeziehungen im Internet,
- C. Informationssicherheit im Internet

betrachtet werden. Zur didaktischen Repräsentation wird das Konzept des Didaktischen Systems auf seine Eignung geprüft und gegebenenfalls für das Didaktische System „Internetworking“ angepasst.

## **1.2 „Medien- und Methodencurriculum“ und Empfehlung der GI**

Ein Konzept zur integrativen Medienbildung wurde 2001 von Brichzin und Stolpmann vorgestellt [BS01]. Dieses so genannte „Medien- und Methodencurriculum“ wurde am Gymnasium Ottobrunn in Bayern für die Sekundarstufe I (Jahrgangsstufen 6 bis 10) erarbeitet und umgesetzt. Ziel des Curriculums ist es, die Verbindlichkeit und Nachhaltigkeit medienbildender Lernziele zu fördern. Dabei wird betont, dass die Methoden nicht nur implizit angewandt sondern explizit reflektiert werden sollen.

Berücksichtigung finden in diesem Konzept insbesondere Massenmedien wie Zeitung und Veröffentlichungen im World Wide Web (WWW). Schwerpunkte liegen in der Beschaffung, der Bewertung und der Präsentation von Information. Informatiksysteme werden aus Anwendersicht als Werkzeuge betrachtet. Ein tieferes Verständnis für Konzepte der Informatiksysteme ist nicht Lernziel. Kenntnisse zu grundlegenden Konzepten von Informatiksystemen werden in der Jahrgangsstufe 6 im Informatikunterricht und in den folgenden Jahrgangsstufen im Mathematikunterricht aufgegriffen.

Die GI hat 1999 in der Empfehlung „Informatische Bildung und Medienerziehung“ [GI99] Ansätze dargestellt, wie Medienerziehung und Informatik im Unterricht miteinander verbunden werden können. Der Unterricht wird dabei aus zwei Perspektiven betrachtet: aus der Sicht der Medienerziehung und aus der Sicht der Informatik. Inhalte mit Medienbezug und informatische Lernziele sind aus beiden Perspektiven voneinander abhängig:

- Medienerzieherische Aspekte können nur dann sachangemessen bearbeitet werden, wenn die informatische Bildung entsprechende Inhalte beisteuert.
- Die informatischen Inhalte werden gerade durch ihren realen medienbezogenen Anwendungskontext für einen allgemeinbildenden Unterricht relevant.

Das heißt zum einen, dass die Bearbeitung von Inhalten mit Medienbezug Kompetenzen aus der Informatik voraussetzt, und zum anderen, dass informatische Lernziele durch den Einsatz von Medien begründet und motiviert werden. Es werden vier Beispiele angeführt, wie die Integration von Inhalten, die durch die Medienerziehung motiviert werden, in den Informatikunterricht denkbar ist. Im Zentrum der Beispiele für den Informatikunterricht stehen typische Informatiksysteme, deren Betrachtung sowohl aus informatischer wie auch medienerzieherischer Sicht möglich ist.

### 1.3 Anforderungen an das didaktische Konzept

Der Ansatz zur Medienbildung von Brichzin und Stolpmann [BS01] weist aus der Sicht der Informatik den Mangel auf, dass Grundlagen der Informatik nicht angemessen berücksichtigt werden. Durch die neue Qualität der internetbasierten Medien ist eine verstärkte Auseinandersetzung mit den zugrunde liegenden Konzepten notwendig. Die GI-Empfehlung führt dazu drei Eigenschaften von Informatiksystemen an: Interaktivität, automatische Verarbeitung von Daten und Vernetzung. Interaktivität bedeutet für Anwender, dass sie zugleich Rezipienten und aktive Teilnehmer sind. Daraus ergeben sich neue Anforderungen an Anwender, die über Kompetenzen wie Lesetechniken und den reflektierten Umgang mit Information hinausgehen. Es geht um eine neue Qualität der aktiven Benutzung. Zudem gibt es bei der Gestaltung von digitalen Medien grundsätzlich mehr bzw. andere Freiheitsgrade. Die automatische Verarbeitung von Daten ermöglicht neue Anwendungen insbesondere abseits der Nutzung als Massenmedium, die neue Fähigkeiten und Kompetenzen nötig machen. Im Hinblick auf die Mensch-Maschine-Interaktion ist dabei ein grundlegendes Verständnis des Informatiksystems erforderlich. Mit Vernetzung als Verbindung von Rechnern sind eine Reihe weiterer Anforderungen durch den möglichen Datenaustausch verbunden, die nicht alleine durch die gegebene Infrastruktur des Internet erfüllt werden. Das Internet weist Sicherheitsrisiken auf, die insbesondere wegen der vielfältigen Anwendungsfelder nicht unberücksichtigt bleiben dürfen. Ein Teil der Verantwortung für die Sicherheit von Information liegt beim Anwender.

Die GI-Empfehlung geht ansatzweise auf solche Aspekte ein. Sie beschreibt jedoch kein Unterrichtsmodell. Wesentliche Merkmale müssen konkretisiert werden. Es muss eine Zielgruppe analysiert und deren Vorwissen und die kognitiven Fähigkeiten berücksichtigt werden. Der Lehr-Lern-Prozess muss erforscht, dargestellt, diskutiert, erprobt und evaluiert werden.

## 2 Vorgehen und Methoden

### 2.1 Strukturierung

In dem in Abschnitt 1.1 vorgestellten Forschungsprojekt geht es darum, Anforderungen an die informatische Bildung zur Förderung der Medienkompetenz von Lernenden zu bestimmen. Das Projekt teilt sich dazu in vier Phasen:

**1. Analyse:** In dieser Phase geht es darum, Alltagserfahrungen der Lernenden zu bestimmen. Die Alltagserfahrungen werden hinsichtlich informatischer Konzepte untersucht, die ein vertieftes Verständnis der damit verbundenen Problemstellungen zulassen. Sie werden als Anknüpfungspunkte im Unterricht und zur Motivation der Lerninhalte genutzt. Zudem sollen geschlechterspezifische Unterschiede durch Gestaltungskriterien für die informatische Bildung berücksichtigt werden.

**2. Angebote:** Im zweiten Teil des Forschungsprojekts werden ein Unterrichtsmodell und

Lernmaterialien entwickelt. Aufbauend auf die Anforderungsanalyse sollen Lehr-Lern-Prozesse gestaltet und evaluiert werden. Die Umsetzung der informatischen Inhalte im Informatikunterricht, die in der ersten Phase bestimmt wurden, ist aus zwei Gründen von Bedeutung: Es geht um den Nachweis, dass es möglich ist, diese Inhalte erfolgreich im Unterricht zu behandeln. Und die Erprobung in der Schulpraxis ist der Ausgangspunkt für die anschließende Evaluation. Unter Berücksichtigung des Didaktischen Systems „Internetworking“ werden Unterrichtssequenzen gestaltet und durchgeführt.

**3. Evaluation:** Zur Bewertung der Akzeptanz durch die Lernenden wird eine schriftliche Befragung durchgeführt. Interviews mit den Stammllehrern der Informatikkurse werden zur Evaluation des Didaktischen Systems heran gezogen. Test-Items, wie sie in den PISA-Tests (Program for International Student Assessment) verwandt wurden, werden dazu genutzt, Aufgabenklassen als Teil des Didaktischen Systems zu validieren und Kompetenzen für Bildungsstandards zu formulieren.

**4. Rückkopplung:** Mit den Ergebnissen aus der Evaluation wird das Didaktische System „Internetworking“ überarbeitet und gegebenenfalls an die Anforderungen angepasst.

## 2.2 Angebote für den Informatikunterricht

Als Grundlage für das Unterrichtsmodell wird das Didaktische System „Internetworking“ entwickelt. Das Didaktische System bietet zudem die Möglichkeit zur Diskussion des didaktischen Konzepts und zum Austausch Lehrender. Die einzelnen Komponenten des Didaktischen Systems sind Wissensstrukturen, Aufgabenklassen und Lernsoftware als besondere Form von Lernmaterialien.

Wissensstrukturen stellen die Lerneinheiten (elementare oder komplexe Fachinhalte) in Relation zueinander. Lerneinheiten können notwendiges Vorwissen sein, das zum Verständnis unter Umständen komplexerer Lerneinheiten notwendig ist. Es kann aber auch sein, dass eine Lerneinheit weitergehenden Lerneinheiten zugute kommt. Die Lerneinheiten werden in einem Und-Oder-Graph dargestellt. Durch eine solche Modellierung ist im Allgemeinen keine Reihenfolge im Unterricht vorgegeben. Es ist dadurch zumeist eine Reihe von Möglichkeiten zur Gestaltung des Lehr-Lern-Prozesses gegeben. Dennoch werden Abhängigkeiten berücksichtigt. Aufgaben lassen sich hinsichtlich Komplexität und Schwierigkeitsgrad, aber auch inhaltlich klassifizieren. Brinda ordnet Aufgabenklassen Lerneinheiten bzw. Fachkonzepten zu [Br04].

Eine wichtige Aufgabe in der didaktischen Aufbereitung für den Unterricht ist es, Lernmaterialien auszuwählen oder zu entwickeln, die Schülertätigkeiten unterstützen. Eine Form von Lernmaterialien beruht auf dem Einsatz digitaler Medien. Der Mehrwert von digitalen Medien liegt insbesondere in der Interaktivität. Das Prinzip der Exploration ist als Grundlage für die Entwicklung von Lernmaterial zum Didaktischen System besonders geeignet. Es lässt zum einen ausreichend Spielraum für die individuelle Gestaltung eines Lehr-Lern-Prozesses in der Schule, setzt aber dennoch didaktische Konzepte konkret im Lernmaterial um.

## **2.3 Bildungsstandards**

Test-Items nach dem PISA-Muster sind unabhängig von einem Curriculum. Sie werden dadurch charakterisiert, dass sie aus einem Stimulus und einer Frage bestehen und der Konzeption ein Kompetenzmodell zugrunde liegt. Der Stimulus stellt den Bezug zu einer Alltagserfahrung der Schülerinnen und Schüler her. Bisher wurden zwei Ansätze für ein Kompetenzmodell in der Informatik publiziert. Der erste Ansatz bestand darin, das Kompetenzmodell der Mathematik mit den Kompetenzklassen Anwendung, Gestaltung und Entscheidung auf die Informatik zu übertragen [Pu04]. Magenheim hat einen Ansatz für ein Kompetenzmodell vorgestellt [Ma05], in dem Kompetenzen nach drei Dimensionen bewertet werden. Die erste Dimension bewertet Kompetenzen hinsichtlich des notwendigen informatischen Systemverständnisses. Die zweite Dimension bewertet den Anwendungsbezug und die dritte Dimension Kompetenzen unter dem Aspekt des Umgangs mit medialen Funktionen eines Informatiksystems.

Schubert schlägt die Verzahnung des Kompetenzmodells von Magenheim mit den Komponenten des Didaktischen Systems vor [Sc05]. Danach korrelieren die Komponenten des Didaktischen Systems mit den Dimensionen des Kompetenzmodells. Notwendige Kompetenzen im Bereich des Systemverständnisses ließen demnach Rückschlüsse auf die Komponente Wissensstrukturen zu. Analog gilt das für den Anwendungsbezug und Aufgabenklassen und für die Nutzung von medialen Funktionen und digitalen Medien als Lernmaterial. Eine mögliche Verzahnung des Didaktischen Systems „Internetworking“ mit einem Kompetenzmodell soll einen Zugang zur Evaluation des Lehr-Lern-Prozesses liefern. In Abhängigkeit der Einordnung von Lernzielen in dieses Kompetenzmodell muss das Unterrichtsmodell entsprechende Schwerpunkte hinsichtlich der Wissensstruktur, der Aufgabenklassen oder dem Einsatz von digitalen Medien als Lernmaterial aufweisen. Zur Evaluation von Lehr-Lern-Prozessen, die durch das Didaktische System begründet werden, ist also zu prüfen, inwieweit das Unterrichtsmodell einen Beitrag zur Entwicklung der Kompetenzen der Lernenden leistet.

Die Entwicklung eines Kompetenzmodells für die Informatik ist noch nicht abgeschlossen. Ein möglicher Ansatz für die Verknüpfung von Didaktischem System und Kompetenzmodell wurde bereits dargestellt. Eine genaue Untersuchung der Verzahnung muss noch erfolgen.

## **3 Entwicklung des Didaktischen Systems „Internetworking“**

### **3.1 Alltagsbezüge**

Im Folgenden wird dargestellt, wie das Vorgehen zur Konstruktion des Didaktischen Systems und der Test-Items aussieht. Ausgangspunkt ist ein Alltagsbeispiel zu einer Internetanwendung. Eine Anforderung an den Informatikunterricht ist schließlich, dass abstrakte Begriffe durch konkrete Beispiele erläutert werden. Ein gutes Beispiel bewirkt, dass eine intuitive Einsicht in die dahinter liegende Abstraktion erhalten bleibt, nachdem Definitionen, Sätze und Beweise längst vergessen wurden. Ein besonderer Typ für ein

Beispiel ist ein Phänomen.

Humbert und Puhmann definieren den Begriff des Phänomens folgendermaßen: „We call the occurrences of informatics [in everyday life and society, S. F.] *phenomena of informatics*“ [HP05, S. 2]. Es lassen sich danach drei Kategorien von Phänomenen unterscheiden: mit direktem Bezug zum Informatiksystem, mit indirekter Kopplung mit einem Informatiksystem und ohne Verbindung zum Informatiksystem. Wobei die letzte Kategorie insoweit zu diskutieren ist, ob ein Bezug zum Algorithmus nicht schon eine Verbindung zum Informatiksystem darstellt. Im Kontext dieses Forschungsprojekts hat die Motivation von informatischen Inhalten durch den Einsatz digitaler Medien im Alltag eine besondere Bedeutung. Solche Phänomene haben einen direkten Bezug zum Informatiksystem und werden der ersten Kategorie zugeordnet.

### 3.2 Lernziele und Aufgabenklassen

In Abbildung 1 sind die Knoten der Wissensstrukturen in einem azyklischen Und-Oder-Graph dargestellt. Dieser Baum bildete den Ausgangspunkt für die Gestaltung einer ersten Unterrichtssequenz zum Thema „Kommunikation in Rechnernetzen“. Die Kanten repräsentieren Relationen wie „ist-hilfreich-für“ oder „ist-notwendig-für“.

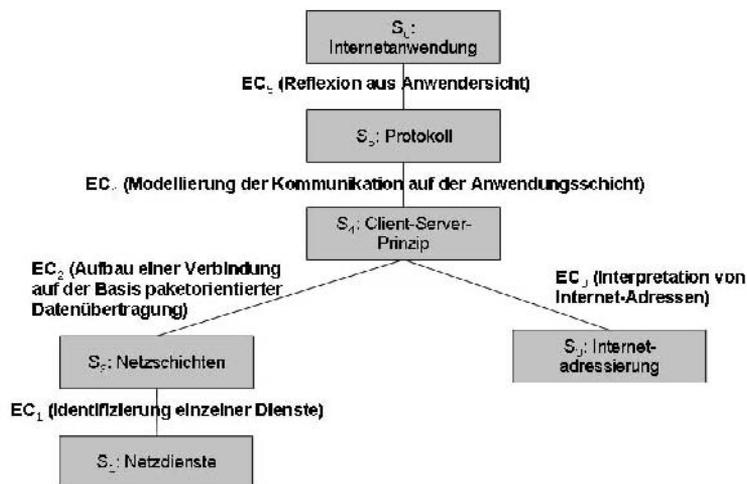


Abbildung 1: Wissensstruktur zu „Kommunikation in Rechnernetzen“

Die einzelnen Groblernziele, repräsentiert durch die Knoten, stehen folgendermaßen in Relation zueinander. Das Thema Netzdienste bietet durch die Metapher der Dienstleistung einen verständlichen Einstieg für die Lernenden. Durch Netzschichten werden die Netzdienste strukturiert. Diese Groblernziele tragen zu einem vertieften Verständnis des Client-Server-Modells bei. Die Internetadressierung wird dafür benötigt, um einzelne Rechner mit den Rollen Client und Server im Rechnernetz zu identifizieren. Erst durch die Adressierung wird eine solche Rollenverteilung möglich. Aufbauend darauf wird die

Rechner-Rechner-Interaktion mit Protokollen beschrieben. Schließlich werden die verschiedenen Lernziele beispielhaft an einer Internetanwendung unter Berücksichtigung wichtiger Aspekte behandelt.

Die Kanten sind mit Aufgabenklassen beschriftet (EC<sub>1</sub> bis EC<sub>5</sub>). Ob es möglich ist, diese Aufgabenklassen tatsächlich zu bestimmen und ob diese Klassen bestimmten Kanten zugeordnet werden können, sind noch offene Forschungsfragen. Aufgabenklassen zur objektorientierten Modellierung wurden dadurch bestimmt, dass vorhandene Aufgaben analysiert und durch die Angaben in der Aufgabenstellung und den Arbeitsauftrag klassifiziert und den Lerneinheiten zugeordnet wurden [Br04]. Wenn es gelingt Aufgabenklassen bestimmten Kanten zuzuordnen, könnte daraus ein Lernpfad konstruiert werden (siehe Abschnitt 4).

Für die Entwicklung des Didaktischen Systems wird das Phänomen Phishing aufgegriffen. Dieses Phänomen gehört zur ersten Kategorie mit direktem Bezug zum Informatiksystem. Anhand dieses Beispiels können die Groblernziele präzisiert werden. In Tabelle 1 sind die Lernziele angeordnet. Die Groblernziele sind die Knoten der Wissensstruktur.

<b>Groblernziele</b>	<b>Feinlernziele</b>
S1: Netzdienste	S11: Bereitstellung zentraler Ressourcen S12: Server als Prozess auf einem Rechner verstehen
S2: Netzsichten	S21: das Konzept von Schichten zur strukturierten Zerlegung beschreiben S22: Anforderungen an Datenübertragung
S3: Internet-Adressierung	S31: Rechneridentität im Internet durch IP-Adresse S32: Komponenten von IP-Adressen (hierarchische Struktur durch Netz und Rechner) S33: Zuordnung einer IP-Adresse zu einer Person
S4: Client-Server-Modell	S41: Verbindungsaufbau zwischen zwei Rechnern S42: Interaktion im Rechnernetz nach dem Client-Server-Modell
S5: Protokolle	S51: Prinzip zur Interaktion zwischen zwei Rechnern S52: Notwendigkeit des korrekten Ablaufs S53: Modellierung mit Zustandsdiagramm
S6: Internetanwendungen	S61: Übertragungsweg einer E-Mail S62: E-Mail-Kopfzeilen verstehen S63: Authentifizierung zum Identitätsnachweis S64: fehlende Authentifizierung beim Simple Mail Transfer Protocol (SMTP)

Tabelle 1: Verfeinerung der Lernziele

Obwohl der Ablauf im Unterricht in aufsteigender Reihenfolge der Groblernziele verläuft, werden die Lernziele ausgehend vom Phänomen Phishing in absteigender Reihenfolge erläutert. Beim Lernen mit der Internetanwendung E-Mail geht es darum, den Ü-

bertragungsweg einer Nachricht zu rekonstruieren. Dazu wird eine E-Mail im Quelltext betrachtet und die Received-Kopfzeilen werden analysiert. Den Lernenden muss dabei bewusst sein, dass bei der Übertragung mit dem Simple Mail Transfer Protocol (SMTP) nicht zwingend eine Authentifizierung der Kommunikationspartner stattfindet. Dazu vergleichen Sie das Post Office Protocol (POP3) mit SMTP anhand der Zustandsdiagramme. Sie lernen Authentifizierung als Identitätsnachweis kennen. Das Thema Protokolle insbesondere des SMTP wird dabei als Prinzip der Interaktion zwischen zwei Rechnern verstanden. Die Notwendigkeit des korrekten Ablaufs wird dabei deutlich. Die Lernenden verschicken dazu E-Mails mit Telnet und SMTP bzw. empfangen diese mit POP3. Zur Modellierung wird ein Zustandsdiagramm genutzt. Als verallgemeinerbares Prinzip zu Interaktionsbeziehungen im Rechnernetz wird die Client-Server-Architektur behandelt. Dabei werden der Ablauf zum Verbindungsaufbau und die anschließende Interaktion mit Abfrage und Antwort thematisiert. Dazu wird das Schulrechnernetz als Beispiel betrachtet, in dem ein zentraler Rechner Ressourcen wie Netzlaufwerke und die Internetzugriffskontrolle durch einen Proxy zur Verfügung stellt. Bei der Betrachtung des Schulrechnernetzes lernen die Schülerinnen und Schüler zudem die Identifikation einzelner Rechner durch IP-Adressen kennen. Dabei unterscheiden sie auch den Netz- und Rechneranteil einer IP-Adresse. Durch den Zugriff auf eine whois-Datenbank über ein WWW-Portal erkennen sie, dass einer IP-Adresse ein Eigentümer zugeordnet werden kann. Das Feinlernziel zu Netzdiensten zur Bereitstellung zentraler Ressourcen wurde bereits im Zusammenhang mit dem Client-Server-Modell besprochen. Die anderen Feinlernziele, die mit den Groblernzielen Netzschichten und Netzdienste verbunden sind, ermöglichen ein vertieftes Verständnis, deren Behandlung ist aber nicht notwendig.

### 3.3 Test-Items nach dem PISA-Muster

Test-Items werden in dem Projekt zur Verifizierung von Aufgabenklassen und dazu genutzt, Kompetenzen für mögliche Bildungsstandards zu formulieren. Puhmann stellt dar, wie die Kompetenzklassen aus der Mathematik in der Informatik angewendet werden können [Pu04]. Es gibt die drei Klassen Anwendung, Gestaltung und Entscheidung. Jedes Test-Item besteht aus einem Stimulus-Material und einer Fragestellung. Das Stimulus-Material beschreibt einen Sachkontext mit Lebensweltbezug. Die Fragestellung zielt darauf ab, bestimmte Kompetenzen zu überprüfen. Im Folgenden werden Beispiele für Test-Items zu den Kompetenzklassen Gestaltung und Entscheidung dargestellt.

Das Stimulus-Material ist eine E-Mail, in der der Absender als kundenservice@musterbank.de angegeben wird. In der Mail wird der Kunde aufgefordert, sein Konto zu reaktivieren. Um diesen Vorgang durchzuführen, ist in der Nachricht ein Link mit der Aufforderung enthalten, auf der Webseite, die darüber erreicht wird, Zugangsdaten und TAN einzugeben.

**Kompetenzklasse Gestaltung:** Ist die Anzeige der Absenderadresse vertrauenswürdig?

- Ja, eine E-Mail kann nur dann versendet werden, wenn sich der Absender erfolgreich authentifiziert hat.
- Nein, es gibt keine sichere Möglichkeit, um zu gewährleisten, dass eine E-Mail

vom angezeigten Absender kommt.

Diese Frage kann richtig beantwortet werden, wenn der Ablauf zwischen E-Mail-Client und -Server bekannt ist – insbesondere die Möglichkeit, dass E-Mails ohne Authentifizierung versendet werden.

**Kompetenzklasse Entscheidung:** Wie könnte man auf die E-Mail reagieren, ohne negative Folgen in Kauf zu nehmen?

- Wenn die Absenderadresse vertrauenswürdig erscheint, sollte man der Aufforderung folge leisten.
- Man sollte die Absenderadresse überprüfen, indem man sich den Übertragungsweg der E-Mail ansieht.
- Man sollte die Webseite der Bank mit einem Web-Browser aufsuchen, indem man die bekannte Adresse von Hand in die Adresszeile des Browsers eingibt.

Zur Beantwortung dieser Frage müssen Lernende eine Entscheidung treffen. Durch die Kenntnis der Möglichkeit, die Information in den Kopfzeilen des E-Mail-Quelltextes zu analysieren, können sie diese Antwort angemessen bewerten. Die dritte Antwort zeigt zudem eine Möglichkeit auf, die dann eingeschätzt werden kann, wenn die grundlegende Vorgehensweise bei einem Phishing-Angriff bekannt ist.

## 4 Fazit und Ausblick

Die untersuchten Arbeiten, die sich mit Medienbildung im Kontext von Informatik beschäftigen sind nicht ausreichend für die Umsetzung in den Informatikunterricht.

Im Rahmen eines durch die DFG geförderten Forschungsprojekts an der Universität Siegen wird das Thema „Informatikunterricht und E-Learning zur aktiven Mitwirkung am digitalen Medienumbruch“ untersucht. Es werden Anforderungen netzbasierter Medien analysiert, Erkenntnisse in einem Unterrichtsmodell umgesetzt und die Ergebnisse empirisch evaluiert.

Als Rahmenkonzept für das Unterrichtsmodell wird das Didaktische System „Internetworking“ entwickelt. Zur Formulierung von Kompetenzen und zur Evaluation werden Test-Items nach dem PISA-Muster erstellt und eingesetzt.

Die nächsten Schritte sind:

- unterrichtliche Erkundung in der Praxis in Kooperation mit zwei Partnerschulen
- Strukturierung der Zusammenhänge zwischen Lernzielen und Aufgabenklassen zur Gestaltung eines Unterrichtsmodells
- Erforschung der Zusammenhänge zwischen Didaktischem System und einem Kompetenzmodell

Zudem wird das Didaktische System an neue Anforderungen angepasst. Untersucht werden soll, ob mit der Zuweisung von Lernzielen anstelle von Lerneinheiten zu Knoten in den Wissensstrukturen und der Interpretation von Aufgabenklassen als Pfade zwischen Knoten eine dynamische Modellierung des Lehr-Lern-Prozesses möglich ist.

## Literatur

- [Br04] Brinda, T. (2004) Didaktisches System für objektorientiertes Modellieren im Informatikunterricht der Sekundarstufe II. Dissertation, Universität Siegen, 2004.
- [BS01] Brichzin, P.; Stolpmann, E. (2001) Medien- und Methodencurriculum Gymnasium Ottonbrunn. In: Keil-Slawik, R.; Magenheimer, J. (Hrsg.): Informatikunterricht und Medienbildung, GI-Edition - Lecture Notes in Informatics, Bonn, S. 63-73.
- [GI99] GI (1999) Informatische Bildung und Medienerziehung. Empfehlung der Gesellschaft für Informatik e. V. (GI).  
URL: [http://www.learn-line.nrw.de/angebote/medienbildung/Foyer/GI/gi\\_empfehlung.pdf](http://www.learn-line.nrw.de/angebote/medienbildung/Foyer/GI/gi_empfehlung.pdf) – geprüft: 27.2.2006
- [HP05] Humbert, L.; Puhlmann, H. (2004) Essential Ingredients of Literacy in Informatics. In: Magenheimer, J.; Schubert, S. (Hrsg.) (2004) Informatics and Student Assessment. GI LNI, Bonn.
- [Ma05] Magenheimer, J. (2005) Towards a Competence Model for Educational Standards of Informatics. In: [Sa05], Documents/452.pdf.
- [Pu04] Puhlmann, H. (2004) Informatische Literalität nach dem PISA-Muster und ihre Operationalisierung durch Test-Items. In: Informatica Didactica 6, 2004.  
URL: <http://www.informatica-didactica.de/InformaticaDidactica/Puhlmann2004.pdf> – geprüft 27.2.2006
- [Sa05] Samways, Brian (Hrsg.) (2005) 8th IFIP World Conference on Computers in Education (WCCE 2005); 4-7 July 2005 - University of Stellenbosch. Cape Town, South Africa: Document Transformation Technologies cc.
- [Sc05] Schubert, S. (2005) From Didactic Systems to Educational Standards. In: Samways, 2005, Documents/397.pdf.
- [Sc93] Schwill, A. (1993) Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 25 1, S. 20-31.  
URL: <http://www.informatikdidaktik.de/Forschung/Schriften/ZDM.pdf> - geprüft am 27.2.2006.
- [SSF05] Schubert, S.; Stechert, P.; Freischlad, S. (2005) Digitaler Medienumbruch. In: LOG IN 25, Heft-Nr. 135, LOG IN Verlag, Berlin, S. 7-8.

# Mobilkommunikation im Informatikunterricht – Thesen und empirische Ergebnisse

Gerrit Kalkbrenner, Benedikt Schultebraucks, Martin Sawatzki

Fachbereich Informatik  
Universität Dortmund  
Otto-Hahn-Str. 16  
44221 Dortmund  
gerrit.kalkbrenner@udo.edu  
benedikt.schultebraucks@udo.edu  
martin.sawatzki@udo.edu

**Abstract:** Das an vielen Schulen verhängte Handy-Verbot vermag eine besondere Motivation für Schüler darstellen, sich eben mit diesem Thema und im Weiteren mit der Mobilkommunikation im Informatik-Unterricht auseinanderzusetzen. Aber auch darüber hinaus bietet dieses Thema Anknüpfungspunkte zu nahezu allen Teilgebieten der Informatik. Die Autoren dieser Publikation sind daher der Frage nachgegangen, in wieweit das Thema Mobilkommunikation als ein holistischer Rahmen für den Informatikunterricht dienen kann. Auf der Basis einer Unterrichtsevaluation konnte der Nachweis erbracht werden, dass auch zentrale Themen der Informatik erfolgreich in einer an der Mobilkommunikation orientierten Unterrichtsreihe vermittelt werden können.

## 1. Einleitung

Der Siegeszug der Mobilkommunikation hat auch vor Schulen nicht halt gemacht. Handys gelten unter Schülern oftmals als Statussymbol. Das vor einigen Jahren an Universitäten groß angelegte Programm zur Einführung von Wireless LAN ist inzwischen auch bis zu den Schulen vorgedrungen. Ein Themenheft der LOG IN [1] hat sich bereits ausführlich mit der Thematik befasst.

Schulen stehen vor einem Dilemma: auf der einen Seite werden mancherorts Handys verboten, da deren Benutzung oftmals im Unterricht störend auffällt, auf der anderen Seite eröffnet aber gerade die Mobilfunktechnik einen breiten Zugang zu vielfältigen Informatik-Themen.

In dieser Publikation wollen wir daher die Frage erörtern, in wieweit das Thema Mobilkommunikation in der Schule anwendbar ist [2]. Erlernen Schüler dabei aber auch Sachverhalte, die zwar nicht unmittelbar die Mobilkommunikation, wohl aber wichtige Bereiche der Informatik betreffen? Die Validierung einer von den Autoren in einer Schule durchgeführten Unterrichtsreihe zur Mobilkommunikation beantwortet etliche Aspekte dieser Fragen [3].

Die aktuelle Diskussion um den ECDL im Informatikunterricht zeigt (vgl. [4]), dass elementare Grundfertigkeiten in der Bedienung von Informatiksystemen weniger im Vordergrund stehen sollen, als grundlegende Informatikkompetenzen. Das bayrische Konzept zur Schulinformatik [5] gibt dementsprechend der objektorientierten Programmierung einen fundamentalen Stellenwert. Die Mobilkommunikation vermag einen Rahmen für den Informatikunterricht darzustellen, anhand dessen zentrale Informatikkompetenzen vermittelt werden (vgl. [6]).

Einige Themenbereiche, wie etwa Funktechnik, sind interdisziplinär. Andere, z.B. Verfahren zur Zuordnung von Frequenzen auf Funkzellen (D- und E-Netz Mobiltelefonie), veranschaulichen Themen der Komplexitätstheorie. Eine besondere Qualität der Thematik Mobilkommunikation ist die Nähe zum Alltagsleben der Schüler und Lehrer. Auf diese Weise werden abstrakte informatische Themen motiviert.

Im Folgenden werden die Inhalte einer Unterrichtsreihe und anschließend die Analyse der Ergebnisse einer begleitenden Evaluation betrachtet.

## **2. Themenspektrum der Mobilkommunikation**

Das Themengebiet Mobilkommunikation fächert sich in vielfältige Unterthemen auf. Von besonderer Relevanz betrachten wir Themen zur Datensicherheit, Datenschutz, Raum-, Zeit- und Ortmultiplexing von Kommunikationskanälen, Landkartenfärbeproblem im Zusammenhang mit der Zuordnung von Sendefrequenzen, Übertragungsprotokolle (GSM, Infrarot, Bluetooth, WLAN, DECT), WAP (Wireless Application Protocol) sowie WML (Wireless Markup Language). Außerdem bieten sich folgende Themen für die interdisziplinäre Zusammenarbeit an: Funktechnik, Beugung und Brechung von Wellen, Mehrwegeausbreitung sowie gesellschaftliche Auswirkungen neuer Technologien und Kommunikationsmedien. Die genannten Themen sind aktuell technisch relevant und genügen weitgehend den Kriterien der fundamentalen Ideen von Schwill [7].

Der Unterricht wird aufgeteilt in mehrere Abschnitte. Nach einer Einleitung in das Themengebiet erfolgt die Vertiefung ausgewählter Aspekte. Den Abschluss bildet ein Projekt, in dem Schüler problem- und handlungsorientiert ein Thema bearbeiten. Insbesondere beinhaltet dieses Projekt Unterricht zu den Masterideen Algorithmisierung, strukturierte Zerlegung und Sprache. Somit ist der allgemein bildende Anspruch erfüllt.

Phasierter Unterricht setzt sich zusammen aus Abschnitten mit fragendentwickelndem Unterrichtsgespräch, Gruppenunterricht und anderen. Von den Schülern werden insbesondere Inhalte erarbeitet, die nicht im direkten Zusammenhang zur Mobilkommunikation stehen. Themen wie die Abschnitte eines Softwareentwicklungsprozesses, objektorientierte Programmierung, Modellierung, Rechnernetze/verteilte Systeme und Teammanagement werden am Rande und bei Bedarf diskutiert.

### 3. Gestaltung der Unterrichtsreihe

Die Unterrichtsreihe wurde als Informatikgrundkurs an einer Gesamtschule in der Jahrgangsstufe 10 erprobt. Der zeitliche Rahmen umfasste ein Halbjahr. In der Unterrichtsreihe werden von den Schülern folgende Inhalte mithilfe der Mobilkommunikation erarbeitet: siehe Tabelle 1 und Fortsetzung in Tabelle 2.

<b>Thema</b>	<b>Inhalt</b>	<b>Unterrichtsgestaltung</b>
Einleitung	Überblick über Mobilkommunikation (IRC5, WLAN, DECT, Bluetooth, Funkmaus, Handy); Kategorisierung nach Reichweite, nach Art: Daten, Telefonie, GPS (Messung der Laufzeit von Signalen), nach Historie (Zeitstrahl): kurzer geschichtlicher Überblick durch Lehrer	fragendentwickelnder Unterricht; Unterrichtsgespräch
Demystifizierung durch technische Hintergründe	Flächendeckende Systeme (Funkzellen, -größe, Zuteilung von Frequenzen, Funkzellen in der Praxis); Bandbreiten und Störempfindlichkeit. Bsp.: Mehrwegausbreitung	fragendentwickelnder Unterricht
Vertiefung der Technologien	IRC5: Belegung, Aufbau, u.ä.; Bluetooth: Anwendungen, Funktion, Sicherheit; WLAN: Aktuelle Produkte/Anwendungen, Aufbaumöglichkeiten von Netzen	Lehervortrag
Datenschutz	Handy-Ortung, Bewegungsprofile	fragendentwickelnder Unterricht
Abhörsicherheit	Verschlüsselung, Kryptografie	fragendentwickelnder Unterricht, Stillarbeit
<b>Projektphase:</b>		
Einführung in WAP	Funktionsweise und Aufbau des WAP, Gateways, Proxys	Lehervortrag
Einführung in WAP	Kategorisierung von WAP-Seiten, realisierbare Inhalte, Speicherung und Übertragung	Expertengruppen
Projektzielplanung; Nachteile des WAP	Anwendungsmöglichkeiten des WAP innerhalb der Schule, schlechte Verbreitung des WAP	gemischte Expertengruppen
Hausaufgabe Analyse	Anforderungen an das Projekt; Gespräch mit potentiellen Kunden	Selbständig geführtes Interview der Schüler

Tabelle 1. Unterrichtsreihe – erste Phase

Aufgrund des zeitlich engen Rahmens und der Unterrichtssituation an der Gesamtschule wurden die Themen und deren Inhalte soweit vertieft, dass sie zur erfolgreichen Bearbeitung der Projekte ausreichten.

<b>Thema</b>	<b>Inhalt</b>	<b>Unterrichtsgestaltung</b>
Projektphasen	Definition der Analyse & Spezifikation des Projekts	Lehrervortrag, Diskussion
Spezifikation	Expertengruppen spezifizieren Anforderungen des Projekts, Modellierung, Erstellung von Artefakten	Expertengruppen
Spezifikation	Diskussion der Gruppenergebnisse	Schülervortrag, Diskussion
Entwurf	WML-Einführung: Funktionsweise, Syntax	Stationenlernen
Entwurf	Modellierung unter Berücksichtigung der technischen Möglichkeiten	Expertengruppen
Entwurf	Diskussion der Modelle	Schülervortrag, Diskussion
Codierung	Umsetzung der Entwürfe & Problembehandlung, Zwischenergebnisse	Expertengruppen, Lehrervortrag
Nachbereitung	Verlauf & Analyse der Projektarbeit	Schülervortrag, Diskussion
Nachbereitung	Präsentation der Projektergebnisse mit Rückmeldung (gegenüber den Kunden)	Schülervortrag

Tabelle 2. Unterrichtsreihe - zweite Phase

## 4. Evaluation zur Unterrichtsreihe

„Evaluation ist das Bemühen, die Qualität von Schule zu verstehen, in der Absicht, sie weiter zu entwickeln.“ [8]

Dementsprechend begleitend zur Durchführung der Unterrichtsreihe erfolgte eine Evaluation, um sowohl den Lernerfolg der Schüler, aber auch die Schultauglichkeit des Themas Mobilkommunikation zu bestimmen.

### 4.1 Zielbestimmung der Evaluation

Um den Erfolg unserer Unterrichtsreihe zur Mobilkommunikation zu erfassen, wählten wir die Methode der Fragebögen. Diese Methodik hat nahezu identische Rahmenbedingungen erlaubt und schaffte damit die Grundlage zur Vergleichbarkeit der Evaluationsergebnisse. Die Identität der Rahmenbedingungen ist bei mehreren Evaluationsschritten, wie bei dieser Untersuchung, unverzichtbar gewesen.

Der Literatur entnehmen wir folgende Empfehlung zur Reihenfolge bei der Durchführung der Evaluation [9]: Ziele, Methodik und Probanden definieren, Überblick über Themenbereich bilden, Ergebnisraum abgrenzen, Evaluation durchführen sowie auswerten.

### 4.2 Evaluationsvorbereitung und Fragestellung

Die Evaluation wurde durchgeführt, um folgende Fragestellungen zu beantworten: Ist

das Thema Mobilkommunikation in der Schule anwendbar? Haben die Schüler etwas zum Thema gelernt? Erlernen die Schüler auch Sachverhalte, die nicht unmittelbar die Mobilkommunikation betreffen? Schließlich: wie kann die Evaluation die zukünftige Qualität des Unterrichts durch Einbezug des Themas nachhaltig verbessern?

Um die Fragen zu beantworten, wurde dreiphasig vorgegangen: ein Prätest bestimmt das vorhandene Vorwissen, ein eingeschobener Test den zwischenzeitlichen Wissenstand und ein abschließender Test ermöglicht Rückschlüsse über den Lernerfolg. Die Probanden werden durch die Schüler verkörpert.

Zur Evaluation wurden Fragebögen entwickelt, die ausgewählte Bereiche der im Praktikum behandelten Mobilkommunikationsthemen und indirekt informatische Grundkenntnisse mithilfe der Mobilkommunikation abfragen. Der Ergebnisraum wurde eingegrenzt, um die Vielfalt von möglichen Antworten bewältigen zu können. Fragen wurden z.B. mit wenigen Antwortmöglichkeiten vorgegeben (Multiple Choice). Darüber hinaus gab es Zuordnungsaufgaben sowie Fragen mit freien Antwortmöglichkeiten.

Auswahl aus dem Fragenkatalog:

- Welche Arten der Mobilkommunikation kennst du? Hierbei gab es keine Vorgaben für die Antwort. Der Proband konnte die ihm bekannten Arten notieren.
- Zu welcher Zeit wurden die ersten Funktelefonnetze in der Bundesrepublik Deutschland angeboten? Mögliche Multiple Choice Antworten waren 1870-1880, 1910-1920, 1950-1960 und 1990-2000.
- In der Entwicklung von WAP Seiten werden, von der ersten Idee bis zur Betrachtung auf dem Handy, verschiedene Phasen durchschritten. Ordne folgende Phasen in zeitlicher Abfolge! (a) Entwurf, (b) Hochladen, (c) Codierung und (d) Analyse. Diese Aufgabe unterscheidet sich von den anderen dadurch, dass hier keine Frage beantwortet wird, sondern durch Zuordnung mehrere Kombinationen möglich sind.

Insgesamt beinhaltet der Fragebogen 11 Fragen mit den o.g. Fragentypen. Bei der Bearbeitung der Bögen blieben die Schüler anonym.

### **4.3 Durchführung der Evaluation**

Nach Abschluss der Vorbereitungen wurden drei Zeitpunkte festgelegt, an denen die Evaluation mit Hilfe der Fragebögen durchgeführt wurde. Dieses Vorgehen ermöglicht eine bessere Aussage insbesondere über die Unterrichtsabschnitte Einleitung und Projekt. Der Inhalt der drei Fragebögen unterschied sich ausschließlich in der Reihenfolge der Aufgaben. Somit konnte sichergestellt werden, dass die Vergleichbarkeit der drei Abschnitte gegeben war. D.h. die Fragen wurden explizit nicht verändert, damit die o.g. Identität der Rahmenbedingungen und der daraus resultierenden Grundlage zur Vergleichbarkeit gegeben ist. Diese Vergleichbarkeit ist nötig um Antworten auf unsere Fragen (s. 4.2) zu bekommen.

Der erste Fragebogen wurde zu Beginn der Unterrichtseinheit bearbeitet. Dabei war den Schülern das Thema des kommenden Unterrichts nicht bekannt. Der zweite Test wurde

nach Durchführung des theoretischen Teils erhoben. Die dritte Evaluationsphase fand zu Beginn der letzten Unterrichtsstunde zur Reihe Mobilkommunikation statt. Zwischen dem zweiten und dritten Fragebogen wurde die Projektphase durchgeführt. Zu betonen ist, dass ein wichtiges Ziel der Projektphase die Vermittlung grundlegender informati-scher Ideen mithilfe der Mobilkommunikation war. Den Nachweis darüber zu erbringen, war u.a. Aufgabe der Auswertung zur Evaluation.

#### **4.4 Auswertung der Evaluation**

Wie oben beschrieben wurden drei Zeitpunkte festgelegt, an denen die Fragebögen verteilt wurden. Der für die Auswertung einfachste Typ ist die Multiple Choice Aufgabe. Die quantitative Erhebung der Ergebnisse kann auf verschiedene Weise erfolgen. So hat sich z.B. bei der Auswertung der 2. Beispielfrage (s.u.) die Möglichkeit einer geordneten Folge angeboten, so dass auch die Größe der Abweichung quantitativ erfasst werden konnte.

Die Aufgabe zur Reihenfolge der Projektphasen hat einen größeren Gestaltungsaufwand gefordert, da bei vier Antwortelementen 24 Kombinationsmöglichkeiten entstanden sind. Das Ziel ist es gewesen sicherzustellen, dass eine Auswahl der Ergebnisse möglichst anschaulich dargestellt wird.

Den höchsten Aufwand bereitete die Auswertung der freien Antworten. Durch das Kategorisieren der gegebenen Antworten nach fachspezifischen Kriterien bezüglich der Frage hat sich sicherstellen lassen, dass auch hier eine anschauliche Darstellung der Ergebnisse möglich ist.

Die Ergebnisse werden im Folgenden auf verschiedene Arten präsentiert. Dabei spielt die statistische Analyse mit ihrer Vorgehensweise eine entscheidende Rolle. Insbesondere wurden die Varianz, der Mittelwert und die relative Häufigkeit berechnet. Im Zuge dessen ist es auch möglich, sich nicht nur auf qualitative Ergebnisse zu beschränken. Die Berücksichtigung der Leistungsverteilung erlaubt ebenso quantitative Ergebnisse.

## **5. Ergebnisse**

Die im Folgenden präsentierten Ergebnisse betreffen den Lernzuwachs der Schüler und die schulische Relevanz des Themas Mobilkommunikation. Ein direkter Vergleich der Ergebnisse ist möglich, da das erste Abbild zur Leistungsverteilung vor Beginn der Unterrichtseinheit erstellt wurde. Bei der Beleuchtung der Ergebnisse beschränken wir uns an dieser Stelle exemplarisch auf drei Punkte, die im Folgenden näher erläutert werden:

### **5.1 Frage: Welche Arten der Mobilkommunikation kennst du?**

Diese Frage forderte freie Antworten, die wie folgt beantwortet wurden:

## Verteilung von Arten der Mobilkommunikation

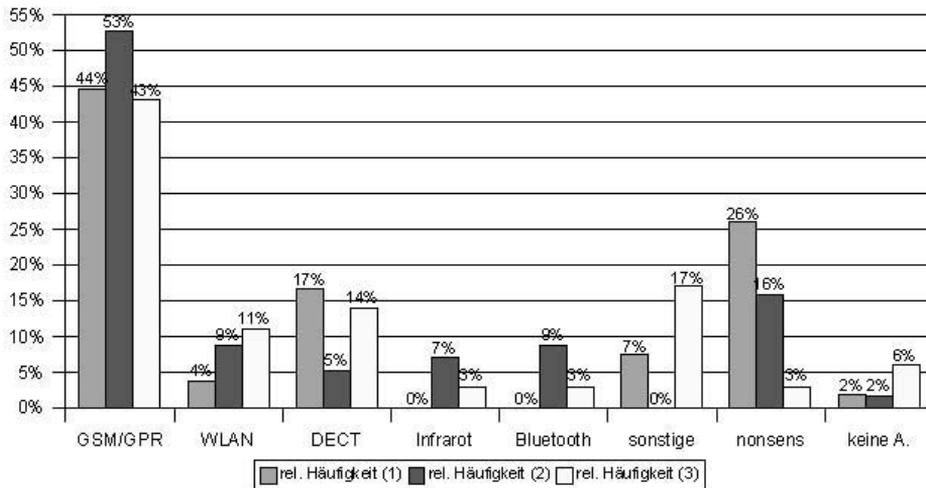


Abbildung 1. Mobilkommunikation – erste Frage

**Ziele:** Einschätzung zur Bedeutung der Mobilkommunikation, Messung des Lernerfolgs der Schüler im Unterricht

**Erwartete Ergebnisse der Auswertung:** Wir erwarteten in der erste Auswertung, dass den meisten Schülern die Kategorien der Mobilkommunikation unbekannt sein würden, dies ist gleichbedeutend mit einer hohen Varianz. In der zweiten Auswertung sollte eine annähernde Gleichverteilung eintreten. Ebenso war zu erwarten, dass die Anzahl der bekannten Arten ansteigt. Wir erhofften uns, dass der dritte Fragebogen gegenüber dem zweiten nur schwache oder keine Veränderungen zeigen würde, da in der zugehörigen Zeitspanne die Projektphase stattfand.

**Interpretation der Auswertung:** Die ersten zwei Erwartungen wurden bestätigt. Die erwartete Ausprägung der Bandbreite an Antworten wurde jedoch nur bei einem geringen Teil der Klasse sichtbar. Daraus ist zu schließen, dass das Ziel der ersten Unterrichtsphase, die Entmystifizierung des Begriffs Mobilkommunikation in der Breite nicht erfolgte. Das Ergebnis des dritten Fragebogens überrascht durch eine unerwartete Entwicklung. Anscheinend trägt gerade das Projekt dazu bei, dass den Schülern viele Sachverhalte insbesondere durch praktisches Arbeiten klarer wurden. Demnach erfolgte die Entmystifizierung des Begriffes Mobilkommunikation in der Projektphase.

## 5.2 Frage: Zu welcher Zeit wurden die ersten Funktelefonnetze in der Bundesrepublik Deutschland angeboten?

Diese multiple choice Frage konnte mit quantitativen Verfahren ausgewertet werden:

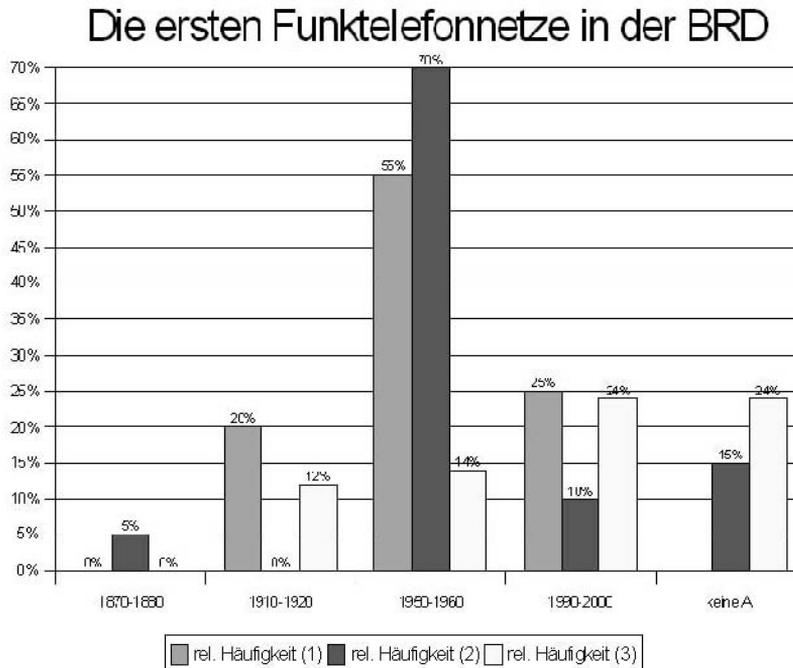


Abbildung 2. Mobilkommunikation – zweite Frage

**Ziele:** Historische Einschätzung zur Entstehung der Mobilkommunikation, Messen des Lernerfolgs der Schüler im Unterricht

**Erwartete Ergebnisse der Auswertung:** Zu Beginn der Unterrichtsreihe wurde eine starke Streuung der zitierten Jahreszahlen erwartet. In den Ergebnissen des zweiten Fragebogens sollte die Streuung, aufgrund der Thematisierung im Unterricht, fallen. Im Vergleich zum zweiten Fragebogen sollte der dritte Fragebogen keine größere Varianz aufweisen.

**Interpretation der Auswertung:** Die erste Auswertung zeigte überraschenderweise eine deutliche Zentrierung, ohne große Streuung oder Ausreißer. Der zweite Fragebogen erbrachte hingegen eine deutliche Steigerung der richtigen Antworten, aber auch eine qualitativ größere Streuung. Die Erwartungen an den dritten Fragebogen wurden nicht erfüllt, da hier eine gegenläufige Entwicklung hin zu größerer Streuung und Abweichung von der richtigen Antwort festgestellt wurde. Die Ergebnisse lassen die Interpretation zu, dass die Jahreszahlen zur Mobilkommunikation in der ersten Unterrichtsphase zwar

verstanden wurden, jedoch auch schnell – vermutlich wegen beigemessener Unwichtigkeit – wieder vergessen wurden.

**5.3 Frage: In der Entwicklung von WAP Seiten werden, von der ersten Idee bis zur Betrachtung auf dem Handy, verschiedene Phasen durchschritten. Ordne folgende Phasen in zeitlicher Abfolge! (a) Entwurf, (b) Hochladen, (c) Codierung und (d) Analyse.**

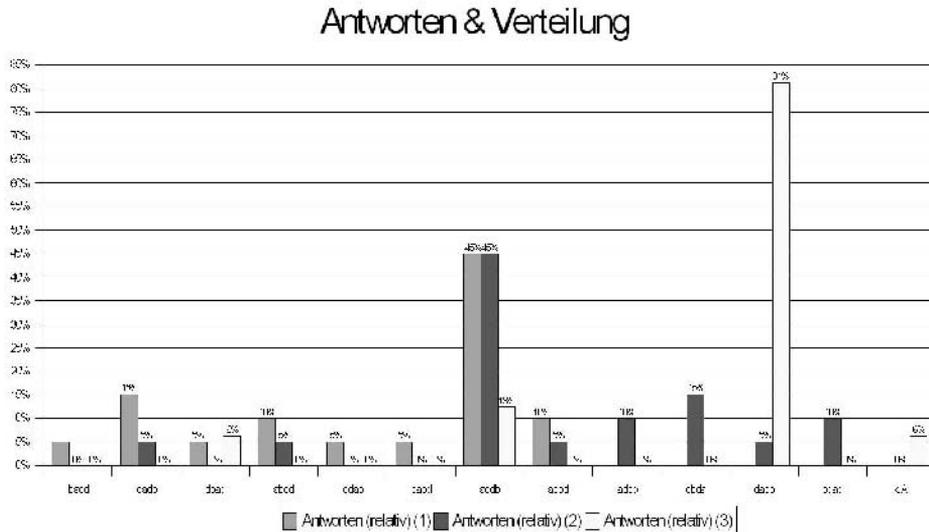


Abbildung 3. Mobilkommunikation – dritte Frage

**Ziele:** Der Proband sollte die Entwicklungsschritte zu einer WAP Seite erlernen. Die Frage beschränkt sich in ihrer Sache jedoch nicht nur auf WAP (vgl. [7]).

**Erwartete Ergebnisse:** Wir erhofften einige Kombinationen aus den 24 Permutationen der Reihenfolge. Der dritte Fragebogen sollte nach der praktischen Erprobung des Themas zu einer geringeren Abweichung von der korrekten Kombination führen.

**Interpretation der Auswertung:** Der erste Fragebogen wies nur ein Drittel der Permutationen auf. Im Vergleich zum zweiten variierte lediglich die Art der Permutationen, nicht jedoch deren Anzahl. Da in dieser Aufgabe auch richtige Teilfolgen bewertet wurden, ließ sich eine Leistungsverteilung abbilden. Diese zeigte eine deutliche Unwissenheit der Schüler. Erst im dritten Fragebogen sank die Anzahl der in den Lösungen vorkommenden Permutationen auf drei. Bemerkenswert ist, dass die Projektphasen nicht Unterrichtsinhalt zum Thema Mobilkommunikation waren. Die Vermittlung von themenfremden Inhalten - in diesem Fall die Projektphasen - gelang hervorragend. Wie man den o.g. Ergebnissen entnehmen kann, eignet sich besonders das Thema Mobilkommunikation als Rahmen oder auch Schafspelz um theoretische Inhalte zu vermitteln.

## 6. Zusammenfassung und Fazit

Die Autoren dieser Publikation sind der Frage nachgegangen, in wieweit das Thema Mobilkommunikation als ein möglicher Rahmen für den Informatikunterricht dienen kann. Daher wurde von den Autoren eine Unterrichtsreihe entwickelt und erprobt, um empirische Aussagen zu den Fragen zu erhalten. In dieser Publikation wurden die Ergebnisse der Erprobung dokumentiert. Die Ergebnisse der Evaluation bestätigen den Erfolg der Unterrichtsreihe und insbesondere ihre Schultauglichkeit.

Die Ergebnisse der Evaluation lassen den Schluss zu, dass auch zentrale Themen der Informatik erfolgreich in einer, an der Mobilkommunikation orientierten, Unterrichtsreihe vermittelt werden können. In den ausgewählten Beispielen konnte der Nachweis erbracht werden, dass die Mobilkommunikation einen geeigneten Rahmen für informatische Themen darstellen kann, die somit den Wolf im Schafspelz darstellen (Schafspelzprinzip).

Eine zukünftige Aufgabe besteht darin, den Nachweis zu erbringen, dass eine Motivationssteigerung und Förderung des Eigeninteresses der Schüler, nicht zuletzt durch die Nähe des Themas Mobilkommunikation zum Alltagsleben, besteht.

Die vollständigen Evaluationsergebnisse sind in [3] nachlesbar.

## Literaturverzeichnis

- [1] "Themenheft zu Mobilkommunikation", LOG IN, Heft Nr. 125, 2003
- [2] Koch, K-P.; Ploeger, M.; Osebold, S.; Kalkbrenner G.: "Schulbuch: Mobilkommunikation im Informatikunterricht", Universität Dortmund 2006
- [3] Schultebrucks, B.; Sawatzki, M.: "Ergebnisse und Praxisbeispiele zur Mobilkommunikation in der Schule", Universität Dortmund 2006
- [4] Micheuz, P.: "Auf dem Weg zu Standards", LOG IN, Nr. 135, 2005
- [5] Hubwieser, P.: "Von der Funktion zum Objekt – Informatik für die Sekundarstufe 1": GI Proceedings P-60, Hrsg.: Friedrich, S.: Unterrichtskonzepte für die informatische Bildung, INFOS 2005, ISBN – 88579-389-X
- [6] Kalkbrenner, G.: "Thema Mobilkommunikation als Rahmen und Motivation für den Informatikunterricht", Universität Dortmund 2006
- [7] Schubert, S.; Schwill, A.: "Didaktik der Informatik", Spektrum Akademischer Verlag 2004
- [8] Schratz, M.: <http://134.96.59.15/SE/Texte/Evaluation%202.htm> (20.10.2006)
- [9] Haenisch, H; Kindervater, C.: "Evaluation der Qualität von Schule und Unterricht", Kettler (Bönen) 1999
- [10] <http://www.learn-line.nrw.de/angebote/schulprogramm/umsetzung/evaluation/index.html>, [www.evaluation-bw.de](http://www.evaluation-bw.de), [www.isb.bayern.de](http://www.isb.bayern.de) (20.10.2006)

# Ein Test- und Analyseverfahren zur Kontrastierung von Problemlöse-Prozessen informatischer Hoch- und Niedrigleister – erste Ergebnisse einer Pilotstudie

Bertold Kujath

Didaktik der Informatik – Universität Potsdam  
August-Bebel-Straße 89  
14482 Potsdam  
kujath@uni-potsdam.de

**Abstract:** Was machen Niedrigleister falsch und Hochleister besser? Problemlöseerfolg bei Informatikaufgaben ist nicht nur das Ergebnis von vorher erworbenem Wissen und Übungspraxis, sondern auch der Aktivitäten, die während der Aufgabenbearbeitung entwickelt werden. Dies gilt besonders für unbekannte Aufgaben, bei denen dem Problemlöser keine vorab bekannten Problemlöseschemata zur Verfügung stehen. Im folgenden wird ein TestszENARIO beschrieben, in dem mit der Methode des Lauten Denkens und anschließender qualitativer und quantitativer Analyse der Verbalisierungen die Problemlösestrategien von Hoch- und Niedrigleistern in der Informatik erfasst, analysiert und verglichen werden können. Erste Ergebnisse der kürzlich abgeschlossenen Pilotstudie werden zur Diskussion gestellt. Es kann davon ausgegangen werden, dass Hochleister ihren Problemlöseerfolg beim Bearbeiten unbekannter Aufgabentypen durch die Wahl effizienterer Problemlöseaktivitäten erreichen und andererseits Niedrigleister über keine oder wenig adäquate Problemlösestrategien verfügen. Das hier vorgestellte TestszENARIO bietet aber auch die Möglichkeit, weiterführende Fragen, z.B. nach den in der Informatik vorherrschenden Denkstilen zu beantworten.

## 1 Einleitung

Die Frage, worin sich die Problemlöseprozesse Hochleistender von denen Niedrigleistender unterscheiden und welche unterschiedlichen Problemlöse- und Denkstrategien hierbei zum Einsatz kommen, soll mit der Methode des *Lauten Denkens* untersucht werden. Bei dieser in der Problemlöseforschung seit langem etablierten, wenn auch kontrovers diskutierten Methode, werden die Probanden gebeten, beim Lösen von Informatikaufgaben alle Gedankengänge laut auszusprechen. Studien zur Validität dieses Datenerhebungsverfahrens belegen, dass diese Art des Problemlösens bei Beachtung gewisser limitierender Faktoren den Problemlöseprozess nicht qualitativ verändert [NeSi72, Deff84, ErSi93].

Versuche mit Lautem Denken laufen in der Problemlöseforschung üblicherweise als sog. *Kontraststudien* ab, bei denen in der Regel zwei Extremgruppen hinsichtlich bestimmter Kriterien miteinander verglichen werden. Diese sind meistens als sog. Experten-Novizen-Vergleiche konzipiert. Waren diese Vergleiche anfangs noch auf formale Bereiche wie Mathematik [Dunc66, Ande81], Physik [Ande00], aber auch z.B. Schach [Ande00] begrenzt, finden sich Experten-Novizen-Vergleiche mittlerweile in nahezu allen Problembereichen bis hin zu Betrachtungen von Umweltproblemen, Geschichte [GrRe97] oder Fragestellungen in der medizinischen Diagnostik [Ande00] wieder. Novizen eines Fachgebiets verfügen im Gegensatz zu Laien zwar über gewisse fachliche Kenntnisse, haben aber – und das unterscheidet sie von Experten – noch keine automatisierten Lösungsprozeduren erworben, die sie zur Bearbeitung von Problemen abrufen können. Das Lösen von domänenspezifischen Aufgaben erfolgt bei Novizen also durch Erarbeitung eines Lösungsweges mehr oder weniger unter Einsatz allgemeiner und daher unspezifischer Problemlösestrategien. Experten haben dagegen durch jahrelange zielgerichtete Übung und Erfahrung einen Fundus an vorgefertigten Problemlöseschemata zur Verfügung, die lediglich an eine abweichende Problemlösesituation angepasst werden müssen [Ande00, Frie01]. Als weiteren Befund der Expertiseforschung berichtet [Spad92] über das vermehrte Auftreten systematischer Fehler bei Novizenlösungen im Gegensatz zu den selteneren zufälligen Fehlern bei Experten.

Allerdings besteht beim Experten-Novizen-Paradigma zwischen den Vergleichsgruppen üblicherweise ein großer Altersunterschied, der die Verwendbarkeit der Ergebnisse für den hier verfolgten Zweck fraglich erscheinen lässt. So hat man zwar herausgefunden, dass Experten Problemlöseschemata abrufen, während Novizen Lösungen erarbeiten müssen [Frie01]. Das Problemlöseverhalten der Experten ist aber von Faktoren geprägt (z.B. umfassende Wissensbasis und jahrelange Problemlösepraxis), die nicht ohne weiteres auf Novizen übertragen werden können. Im hier vorliegenden Ansatz sollen daher *Kontrastgruppen gleichen Alters* untersucht werden, die sich in ihrer Problemlöseleistung deutlich unterscheiden. Um eine vergleichbare Ausgangssituation herzustellen, sollen beide Gruppen während des Erarbeitens von Lösungswegen unbekannter Aufgabentypen beobachtet werden. Da diese Aufgabentypen in der Regel weder wissens- noch könnensbasiert gelöst werden, zielen sie auf Unterschiede in den kognitiven Fähigkeiten ab. Dies soll auch durch den Vergleich der Lösungen mit den Ergebnissen eines Intelligenzstrukturtests verdeutlicht werden.

Über den Problemlöseerfolg entscheidet neben Intelligenz, Kreativität oder Motivation auch der bei der Aufgabenbearbeitung vorherrschende Denkstil, also die Art, wie jemand seine kognitiven Mittel einsetzt [StZh01]. Der Denkstil entscheidet mitunter sogar über den Lernerfolg im Unterricht. So berichtet [Borr04], dass Schüler oft deutlich mehr Schwierigkeiten im Mathematikunterricht haben, wenn sie andere mathematische Denkstile favorisieren als die Lehrkraft. Mit den Möglichkeiten der qualitativen Datenanalyse soll daher hinterfragt werden, ob sich in der Informatik ebenfalls unterschiedliche Denkstile identifizieren lassen.

Sollten die Untersuchungen ergeben, dass charakteristische Problemlösestrategien der Hochleister ursächlich für den höheren Problemlöseerfolg und als Strategiemuster beschreibbar sind, kann in einem weiteren Schritt geprüft werden, inwieweit diese Strate-

gien an Niedrigleister vermittelt werden können.

Unter den Vorarbeiten anderer Autoren zu ähnlichen Fragestellungen sind besonders die folgenden zu erwähnen: R. Borromeo Ferri [Borr04] forschte zu mathematischen Denkstilen, indem sie 12 Schüler der 6. und 9. Jahrgangsstufe beim paarweisen Lösen mathematischer Aufgaben beobachtete. Sie identifizierte bei ihnen drei bereits von [Burt97] bei praktizierenden Mathematiklehrern und –lehrerinnen beschriebene mathematische Denkstile durch qualitative Datenanalyse. Da in unserer Studie ein direkter interindividueller Vergleich angestrebt ist, werden die Versuche in Einzelsitzungen durchgeführt.

G. Friege [Frie01] führte Experten-Novizen-Vergleiche beim Lösen elektrotechnischer Aufgaben durch. Die Auswertung der Problemlöseprozesse basierte auf den schriftlichen Lösungswegen der Versuchspersonen, die Ergebnisse wurden mit dem Berliner Intelligenzstrukturtest (BIS) abgeglichen. Vorangegangen waren Voruntersuchungen mit Teilnehmern der Physik-Olympiade. In unserer Studie sollen unmittelbar an die Bearbeitung der Probleme anschließende retrospektive Interviews zum Problemlöseerleben durchgeführt werden, um zusätzliche Information über die individuellen Strategien und Entscheidungsgründe zu gewinnen.

## **2 Hypothesen**

Erfolgreiche, d.h. schnelle und effiziente Problemlöser lösen Informatikaufgaben mit adäquaten informatischen Methoden, Niedrigleister tun dies nicht oder nicht in ausreichendem Maße. In Anlehnung an die Fundamentalen Ideen der Informatik [ScSc04] kann man also in den Problemlöseprozessen informatischer Hochleister Prinzipien des Algorithmisierens, des strukturierten Zerlegens und sprachlicher Konzepte erwarten. Ebenso sind die Methoden des allgemeinen Problemlösens wie Zerlegen in Teilprobleme, planvolles Hypothesenbilden und –testen bei Hochleistern besser ausgeprägt. Als Konsequenz daraus sind die Problemlösestrategien von informatischen Hochleistern effizienter, schneller und führen häufiger zu richtigen Ergebnissen. Bei der Analyse von Verbalprotokollen werden sich daher verstärkt Kategorien des allgemeinen und des informatikspezifischen Problemlösens identifizieren lassen.

Ein weiterführender Forschungsaspekt ist die Analyse der Verbalisierungen nach den in der Informatik vorherrschenden Denkstilen. Ähnlich wie in [Borr04] für mathematische Denkstile beschrieben, wird erwartet, dass sich auch in der Informatik verschiedene für die Informatik spezifische Denkstile identifizieren lassen. Es soll untersucht werden, inwieweit sich diese von den mathematischen Denkstilen abgrenzen lassen.

## **3 Probandenauswahl**

Für die hier beschriebene Studie wurden die Probanden für die Gruppe der Hochleister aus den Endrundenteilnehmern des Bundeswettbewerbs Informatik 2005 (BWInf 05)

angeworben. Insgesamt 23 dieser Endrundenteilnehmer nahmen im Herbst 2005 an einem Intelligenzstrukturtest teil, bei 15 Teilnehmern wurde ein IQ von >130 ermittelt, was allgemein als Schwellenwert für eine psychometrische Hochbegabung angesehen wird [WaWe90]. Da alle diese Testteilnehmer durch Erreichen der Endrunde des Bundeswettbewerbs offenbar zugleich über eine Leistungsexzellenz im Fach Informatik verfügen, erscheinen sie als besonders geeignet für die Gruppe der Hochleister in dieser Studie.

Die Gruppe der Niedrigleister soll aus denjenigen Gymnasiasten des Landes Brandenburg ausgewählt werden, die in der gleichen Jahrgangsstufe wie die Endrundenteilnehmer des BWInf 05 sind, ebenso wie diese Informatik als Leistungskurs haben, aber schlechte Schulleistungen in diesem Fach erbringen.

Üblich ist bei solchen Versuchen, vor der eigentlichen Testserie das Versuchsszenario durch sog. *Pilotstudien* zu überprüfen. An den zwei Testdurchläufen der hier beschriebenen Pilotstudie nahmen bisher insgesamt 6 wissenschaftliche Mitarbeiter und studentische Hilfskräfte des Instituts für Informatik der Universität Potsdam teil.

## **4 Versuchsablauf**

### **4.1 Datenerhebungsverfahren**

Grundlage der Datenerhebung ist eine Kontraststudie mit gleichaltrigen Hoch- und Niedrigleistern. Beide Gruppen erhalten Aufgaben im Stile von nicht unbedingt alltäglichen Klausur- und Übungsaufgaben aus Lehrbüchern der Informatik und sollen Lösungen mithilfe von Lautem Denken erarbeiten. Da Lautes Denken aber nur diejenigen Gedächtnisinhalte erfasst, die dem Probanden tatsächlich bewusst sind, werden anders als bei Versuchen dieser Art mit Experten insbesondere metakognitive Prozesse, also Vorgänge zur Steuerung und Überwachung der Problemlöseprozesse, ausgeblendet. Eine retrospektive Befragung der Probanden direkt nach jeder Aufgabe soll daher auch diejenigen Daten erfassen, die durch Lautes Denken nicht gewonnen werden können. Dies ermöglicht Aussagen über den Denkstil, die individuellen Entscheidungsgründe und eventuelles Vorwissen der Probanden.

Die Auswahl der Informatikaufgaben erfolgte nach folgenden Kriterien. Die Aufgaben sollen

- informatische Inhalte repräsentieren,
- möglichst unbekannt sein,
- einen gewissen Schwierigkeitsgrad aufweisen und auch für Hochleister noch anspruchsvoll sein,
- bei ihrer Bearbeitung möglichst wenig Fachwissen erfordern, da dieses interindividuell unterschiedlich ausgeprägt sein und somit die Problemlösestrategien überlagern kann.

Von 15 in der Pilotphase getesteten Aufgaben wird im folgenden eine Aufgabe mit den zugehörigen Analyseergebnissen exemplarisch vorgestellt. In diesem als Aufgabe 1 bezeichneten Problem sollten die Probanden das Dekrementieren einer Binärzahl um 1 auf einer Turingmaschine implementieren. Bei der Analyse wurde besonderes Augenmerk auf die Phase gerichtet, in welcher der Algorithmus erarbeitet wurde.

Alle Probanden hatten einen Schreibblock mit Stift zur Verfügung. Die während der Aufgabenbearbeitung angefertigten Skizzen wurden ebenso wie die Verbalisierungen der Versuchspersonen mit einer Videokamera aufgezeichnet.

## 4.2 Datenauswertungsverfahren

Das Videomaterial wurde zunächst Wort für Wort transkribiert, die von den Probanden angefertigten Notizen und Skizzen eingescannt und als Bild den entsprechenden Textpassagen zugeordnet. Zur Dokumentation des zeitlichen Verlaufs wurden die Protokolle mit einem fortlaufenden Zeitstempel versehen. Im Originaltext wurden erste Auffälligkeiten mit Memos kommentiert. Für weitere Analyseschritte und zur späteren grafischen Darstellung der Problemlöseprozesse wurde der Text nach den sog. *Z-Regeln* nach Mayring [Mayr00] semantikerhaltend in eine Kurzform transformiert. Bei der Analyse der Verbaldaten wurde auf Verfahren der qualitativen und quantitativen Datenanalyse zurückgegriffen, wie sie etwa in [ErSi93, Chi97, GILa04, GlSt98, Mayr00, Stäu81] beschrieben werden. Im Anschluss daran wurden auffällige Textpassagen in den Verbalprotokollen in ex-ante-Problemlösekatogorien eingeordnet, die im folgenden Abschnitt erläutert werden.

## 5 Problemlösekatogorien

Durch theoretische Vorüberlegungen wurden zunächst Problemlösekatogorien erarbeitet, die durch empirisch aus der Pilotstudie abgeleitete Erkenntnisse ergänzt und verfeinert werden konnten. Diese sollen später induktiv durch spezielle in den Verbalprotokollen identifizierte Katogorien erweitert werden. Das hier vorgeschlagene hierarchische und vernetzte Katogoriensystem unterscheidet drei Klassen von Katogorien:

- I. Katogorien der *allgemeinen Problemlösetheorie* [Funk03, MaFr92, Spad92, MüPr02], z.B. einzelne Problemlöse-Prozessphasen oder allgemeine Heuristiken.
- II. *Informatikspezifische Problemlösekatogorien*, das sind im wesentlichen die fundamentalen Ideen der Informatik [ScSc04] wie z.B. divide-and-conquer-Verfahren, Rekursion, strukturierte Zerlegung oder auch fachspezifische Heuristiken, fachspezifisches Vorwissen oder informatische Elemente in den Skizzen der Probanden.
- III. *Aufgabenspezifische Problemlösekatogorien*, die aus den einzelnen Teil- oder Zwischenlösungen der jeweiligen Probleme, problemspezifischen Schlüsselerkenntnissen oder der Lösung selbst bestehen. Auch aufgabentypische Fehlannahmen gehören

hierzu.

Subkategorien erben Attribute und ihre dimensionalen Ausprägungen von ihren Oberkategorien und können zusätzlich davon abweichende eigene Attribute aufweisen.

Analog hierzu kann auch die Kategorie Versuchsleiterhinweis in drei Klassen mit aufsteigendem Informationsgehalt unterteilt werden:

- I. Hinweise zu *allgemeinen Problemlösestrategien*; Beispiel: „Kann man das Problem vereinfachen?“, „Kennen Sie ein ähnliches Problem?“, „Haben Sie irgendeine Vermutung?“ usw.
- II. Hinweise zu *informatikspezifischen Problemlösestrategien*; Beispiel: „Können Sie hier zunächst einmal globale Aussagen machen und diese dann schrittweise präzisieren?“, „Kann man den Ergebnisbereich in irgendeiner Form strukturieren?“ usw.
- III. *Aufgabenspezifische Problemlösehinweise*; Beispiel: „Versuchen Sie es mit Dreieckszahlen!“, „Das erste Beispiel selbstbezogener Zahlenfolgen ist: 1,2,1,0.“ usw.

Entsprechende Hinweise sollten speziell für jede Aufgabe ausgearbeitet werden. Idealerweise erhalten Probanden in Sackgassensituationen zuerst Hinweise der Klasse I, danach, wenn notwendig, der Klasse II und zuletzt der Klasse III.

So stellt sich beispielsweise die Oberkategorie der allgemeinen Problemlösestrategie wie folgt dar:

Kategorie	Attribute	Dimension
<b>Allgemeine Problemlösestrategie</b> <u>Subkategorien:</u> Hypothese Heuristik Zerlegung in Teilprobleme Analogiebildung Umdenken	<i>Eignung</i>  <i>Effizienz</i>  <i>Flexibilität</i>	gut ↔ schlecht  hoch ↔ niedrig  hoch ↔ niedrig
<b>Beschreibung:</b> „Allgemeine Problemlösestrategie“ ist die Oberkategorie für die in der allgemeinen Problemlösetheorie beschriebenen Phänomene. Dazu gehören das planvolle Aufstellen und Prüfen von Hypothesen nach Dewey (hier als Hypothese zusammengefasst), das Zerlegen eines Problems in Teilprobleme nach Descartes und das Heranziehen von Analogien aus bekannten Lösungen ähnlicher Aufgaben. Außerdem lässt sich der Problemlöseprozess in verschiedene Problemlösephasen unterteilen.		

Weitere Beispiele sind als Auszug aus dem bisher erarbeiteten Kategoriensystem im Anhang aufgeführt.

## 6 Bisherige Ergebnisse der Pilotstudie

Bisher liegen nur Ergebnisse aus der Pilotphase vor; Ergebnisse aus der Hauptstudie sind ab Herbst 2006 zu erwarten. Die Pilotstudie diente in erster Linie der Überprüfung und Evaluation des Testszenarios. Exemplarisch für die Vorgehensweise bei der Analyse der Verbalprotokolle werden hier die Ergebnisse der Aufgabe 1 vorgestellt. Sämtlichen Probanden wurde die folgende Aufgabe präsentiert, die als Kernaspekt die Dekrementierung einer binär-kodierten natürlichen Zahl um 1 auf einer deterministischen Turingmaschine beinhaltet:

„Geben Sie eine deterministische Einband-Turingmaschine  $\tau$  an, die die Funktion:

$$f: \mathbb{N}_0 \rightarrow \mathbb{N}_0, f(n) = \max(0, n-1)$$

berechnet!“

Direkt nach dem Vorlesen der Aufgabe durch den Versuchsleiter wurde ein Startsignal gegeben und mit der Videoaufzeichnung begonnen. Die qualitativen und quantitativen Auswertungen der einzelnen Videosequenzen werden im folgenden für jeden Probanden einzeln skizziert.

### 6.1 Qualitative Analyse

#### Proband G

**Problembearbeitung:** G interessiert sich zunächst für das Kernproblem: Finden des Algorithmus zur Dekrementierung einer Binärzahl um 1, alle anderen Aspekte werden untergeordnet. Das Finden der Regel für die Übertragsbehandlung fällt G nicht ganz leicht. Relativ frühzeitig wird die Lösungsfindung von der Betrachtung einer Turingmaschine abstrahiert.

Diese Versuchsperson löst die Testaufgabe eigenständig und richtig, die anfängliche Frage an den Versuchsleiter bezieht sich ausschließlich auf das Aufgabenverständnis. Sein Problemlöseverhalten entspricht voll und ganz den Erkenntnissen der allgemeinen Problemlöseforschung: engagierte Herangehensweise, Aufteilung in Teilprobleme, planvolles Hypothesenbilden, empirisches und rationales Hypothesenprüfen, Strategiewechsel bei Sackgassensituationen, Anwenden von Heuristiken (Induktion und Extremwertüberprüfung) sowie Umrepräsentieren eines Problems (Addition statt Subtraktion). Dies führt dann auch zur Lösungsfindung. Er zeigt nahezu alle in der Literatur als Metaheuristiken dokumentierten Verhaltensweisen.

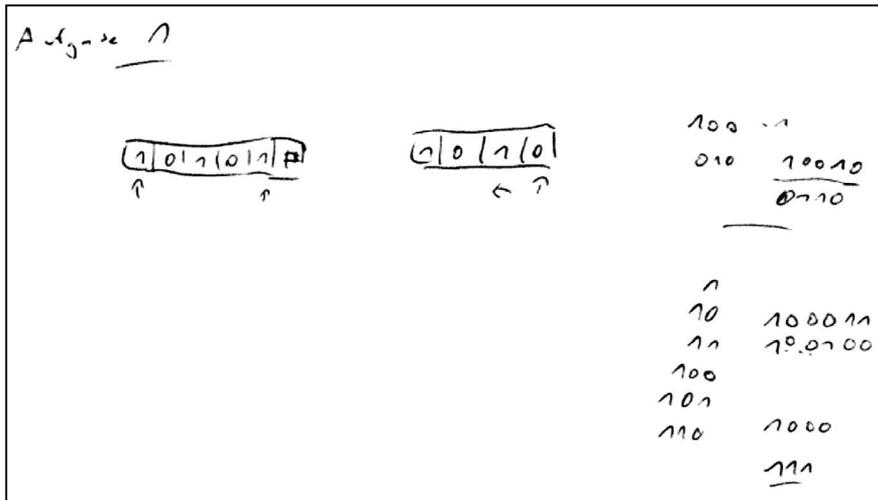
G geht bei der Suche nach einer allgemeinen Dekrementierungsregel auf eine Art vor, die hier als vertikal beschrieben und damit von einer als horizontal bezeichneten Vorgehensweise anderer Probanden unterschieden werden soll. Wie aus Abbildung 1 ersichtlich, schreibt G zunächst sechs kürzere, später rechts daneben zwei längere Binärzahlen aufsteigend untereinander. Er versucht durch direkten Vergleich aufeinanderfolgender Zahlen eine allgemeine Dekrementierungsregel zu finden.

Proband G verzichtet auf die Erstellung einer Turingtafel und erklärt die Aufgabe nach

der informalen Formulierung des Algorithmus für beendet.

**Skizzen:** Skizzen werden von G anfangs zum Verständnis (Selbsterklärung) der Funktionsweise einer Turingmaschine, später dann sowohl zur Hypothesengenerierung als auch zur Hypothesenprüfung eingesetzt.

**Vermuteter Denkstil:** tendenziell analytisch, reflexiv.



**Abbildung 1:** Skizze von G zur Bearbeitung der Aufgabe 1

## Proband K

**Problembearbeitung:** K löst die Aufgabe völlig eigenständig und ohne Intervention des Versuchsleiters, verzichtet aber gänzlich auf eine Evaluation der Lösung. So ist die präsentierte Turingtafel formal gesehen ohne Endzustand, intendiert ist aber das Richtige. Die Vorgehensweise von K ist mit der eines Experten vergleichbar: intensive Analyse der Aufgabe bei gleichzeitiger Repräsentation, zielgerichtete und weitgehend sackgasenfreie effiziente Problembearbeitung. Der Lösungsalgorithmus wird ohne Skizzen durch mentales horizontales Scannen einer vorgestellten Zahl von rechts nach links entwickelt. An jeder Stelle erfolgt die Fallunterscheidung „aktuelle Stelle=1“ oder „aktuelle Stelle=0“, eine Vorgehensweise, die der Arbeitsweise der Turingmaschine entspricht. K verfügt über praktische Erfahrungen mit Turingmaschinen. Das zeigt u. a. sein später wieder verworfener Versuch, bewusst eine Lösung zu präsentieren, die über die Aufgabenstellung hinausgeht, nämlich den Schreib-Lese-Kopf am Ende wieder auf die erste Stelle der Bandinschrift zu positionieren. Insgesamt weist K eine sehr effiziente, deutlich von Merkmalen des Expertendenkens geprägte Problemlösestrategie auf.

**Skizzen:** Die einzige Skizze, die K anfertigt, ist die Dokumentation der Lösung selbst, die sukzessive während des Problemlöseprozesses vervollständigt und somit als eine Art

Leitfaden für den Problemlöseprozess dient.

**Vermuteter Denkstil:** analytisch, anschauungsgebunden.

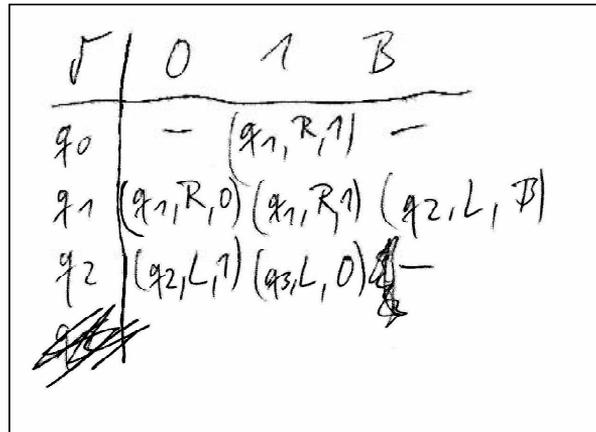


Abbildung 2: Skizze von K zur Bearbeitung der Aufgabe 1

### Proband M

**Problembearbeitung:** die Aufgabenbearbeitung von M ist anfänglich durch ein hohes Maß an Ratlosigkeit gekennzeichnet. Erst nach der Frage des Versuchsleiters, ob man die Aufgabe in Teilprobleme zerlegen (Hinweisklasse I) und wie man die Dekrementierung einer Binärzahl algorithmisieren könne (Hinweisklasse II), kommt ein weitgehend eigenständiger Lösungsprozess in Gang.

Eine spontane Lösungsidee von M ist, die Binärzahl vom Turingband in Stabnotation umzuwandeln, da ihm diese nach eigenen Angaben im Zusammenhang mit Turingmaschinen erheblich geläufiger ist. Er versucht also, auf bekannte Lösungsschemata und Repräsentationen zurückzugreifen, verwirft diesen Ansatz aber wieder.

Im Gegensatz zu G und Q wird der Dekrementierungsalgorithmus nicht anhand mehrerer aufsteigender Binärzahlen gesucht, sondern wie bei Proband K durch Vorstellung einer fiktiven Zahl und Durchlaufen derselben von hinten nach vorne. An jeder Stelle erfolgt dann die Fallunterscheidung, ob eine 1 oder eine 0 gelesen wird mit der entsprechenden Behandlungskonsequenz.

Nach der formalen Notation des Dekrementierungsalgorithmus entwickelt M die Abfrage, ob anfänglich nur Nullen auf dem Band stehen und fügt dies als Programmmodul am Anfang ein. Am Ende der Aufgabenbearbeitung erfolgt die Lösungsevaluation nur anhand der Beispielszahl 1010, ein zufälliger Fehler im Abfragemodul bleibt daher unentdeckt.

**Skizzen:** eine einzige Skizze reicht M zur Hypothesenüberprüfung, die Herleitung einer allgemeinen Dekrementierungsregel erfolgt rein mental. Die restliche Skizze dient der Lösungsnotation.

**Vermuteter Denkstil:** tendenziell analytisch

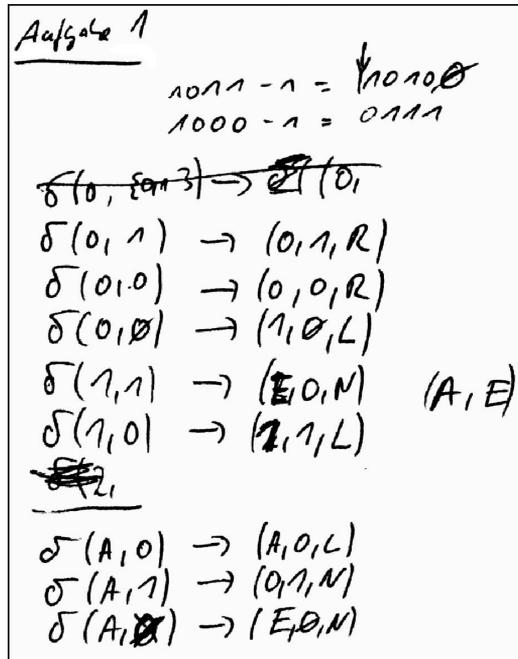


Abbildung 3: Skizze von M zur Bearbeitung der Aufgabe 1

### Proband Q

**Problembearbeitung:** Q geht – wie eher für Novizen typisch – sehr früh zur Problemlösephase über. Er bearbeitet Teillösungen streng sequentiell nach der Reihenfolge ihres Auftretens. Sein erster Fehler erfolgt gleich zu Beginn: Dekrementierung einer Binärzahl von links her. Als der Fehler erkannt wird, bearbeitet Q die Zahl von rechts, scheint aber nach wie vor von links zu denken. Die Implementierung des Lösungsalgorithmus ist unvollständig und auf seine selbstgewählte Beispielzahl 110 zugeschnitten: korrekt ist das Bewegen des Zeigers ganz nach rechts und die Dekrementierung einer Binärzahl, die auf 1 endet. Endet eine Binärzahl auf 0, werden nur diejenigen Zahlen korrekt berechnet, die auf 10 enden. Bei Zahlen mit mehr als einer Null am Ende, wie z.B. 100 oder 1010000, wird zunächst die erste 1 von rechts in eine 0 umgewandelt. Danach wird aber nur noch die 0 unmittelbar rechts daneben in eine 1 umgewandelt und nicht, wie es nötig wäre, alle Nullen bis zum rechten Ende. So wird beispielsweise  $100-1=010$  oder  $1010000-1=1001000$  usw. ausgeführt. Es erfolgt auch keine Evaluation der Lösung mit anderen Beispielzahlen, sonst wäre der Fehler aufgefallen. Jede gefundene Teillösung wird ohne Betrachtung des Gesamtzusammenhangs sofort als Anweisung in die Turing-

tabelle eingetragen. Der Sonderfall  $n=0$  wird überhaupt nicht betrachtet.

**Skizzen:** die erste Skizze ist, wie aus Abbildung 4 ersichtlich, ein Modell einer Turingmaschine mit konkreter Beispielzahl (#100##). Die zweite Skizze folgt dann mit Binärzahlen; zu diesem Zeitpunkt liegt eine korrekte Repräsentation des Problems und eine folgerichtige Informationsbeschaffung vor. Skizzen werden überwiegend als externer Speicher und nur ansatzweise zur Regelherleitung verwendet. Keine einzige Skizze dient zur Hypothesenprüfung, die auch generell nicht stattfindet.

**Vermuteter Denkstil:** eher intuitiv und impulsiv.

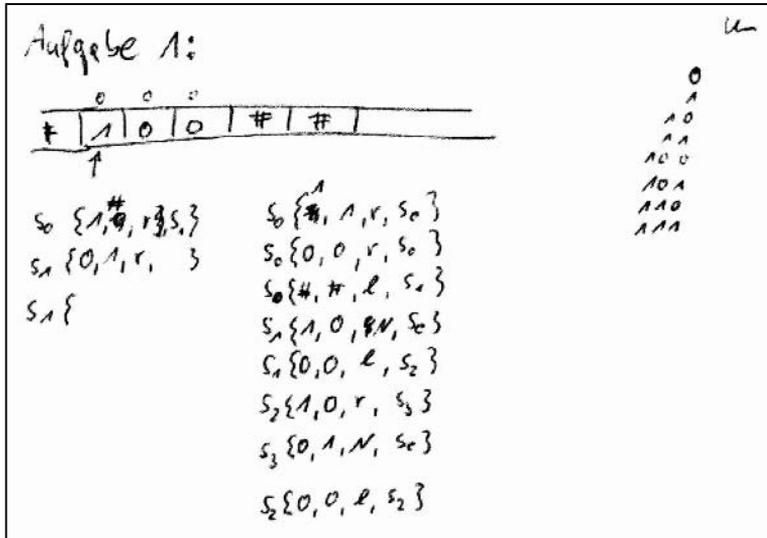


Abbildung 4: Skizze von Q zur Bearbeitung der Aufgabe 1

## 6.2 Quantitative Analyse

Die oben geschilderten Ergebnisse der qualitativen Analyse werden nun durch quantitative Betrachtungen ergänzt. Diese beziehen sich hauptsächlich auf interindividuelle Zeit-Vergleiche der einzelnen Lösungsphasen (Tabelle 1) und die Häufigkeiten bestimmter Phänomene wie beispielsweise Anzahl der Versuchsleiterinterventionen oder Häufigkeit und Art der Skizzen (Tabelle 2). Die Zeiten der Teillösungen beziehen sich auf das erstmalige Bearbeiten und Formulieren der betreffenden Teillösung, unabhängig davon, ob dies durch informale Aussagen oder gleich formal als Übergangsfunktion erfolgte. Die einzelnen Teillösungen der Aufgabe 1 sind im Anhang unter „Klasse III: Aufgabenspezifische Problemlösekatogorien“ beschrieben.

	Proband G		Proband K		Proband M		Proband Q	
<b>Zeiten</b>	Start	Dauer	Start	Dauer	Start	Dauer	Start	Dauer
<b>Gesamtprozess</b>	0:00	8:48	0:00	6:37	0:00	19:22	0:00	10:03
<b>Aufgabenverständnis</b>	0:00	1:28	0:00	2:22	0:00	3:00	0:00	1:26
<b>Repräsentation</b>	1:28	0:32	0:00	2:22	3:00	1:04	1:26	0:50
<b>Problemlösephase</b>	2:00	5:42	2:22	4:15	4:18	12:49	2:56	7:07
<b>Evaluation</b>	7:42	0:45	∅	0:00	17:20	2:02	∅	0:00
<b>Teillösung 1</b>	6:14	0:01	3:51	0:15	15:41	0:20	∅	0:00
<b>Teillösung 2</b>	2:00	0:18	4:55	0:38	9:55	0:03	5:18	0:59
<b>Teillösung 3</b>	2:18	0:07	5:38	0:30	4:36	0:18	6:17	1:05
<b>Teillösung 4</b>	2:32	5:17	5:38	0:30	5:21	0:36	7:22	2:41

**Tabelle 1:** Ergebnis der Zeit-Analyse bei Aufgabe 1

	Proband G	Proband K	Proband M	Proband Q
<b>Skizzen</b>				
<b>Skizzen insgesamt</b>	<b>7</b>	<b>1</b>	<b>3</b>	<b>3</b>
Modellskizzen	2	0	0	1
Hypothesengenerierung	3	0	0	1
Hypothesenprüfung	2	0	1	0
externer Speicher	0	1	2	1
<b>Sonstiges</b>				
<b>VI-Interventionen</b>	<b>1</b>	<b>0</b>	<b>16</b>	<b>4</b>
davon aktiv	0	0	8	2
durch Frage an VI	1	0	8	2
<b>richtige Fachtermini</b>	<b>6</b>	<b>10</b>	<b>7</b>	<b>4</b>

**Tabelle 2:** Frequenzanalytische Ergebnisse bei Aufgabe 1

## 7 Diskussion

Die Ergebnisse der Pilotstudie sind aufgrund der niedrigen Probanden- und Aufgaben-zahl für verallgemeinernde Aussagen zur Zeit noch ungeeignet. Zumindest aber kann festgestellt werden, dass die vorab aufgestellten Vermutungen durch die Versuchsergeb-nisse nicht widerlegt wurden, die Ergebnisse der betrachteten Einzelfälle waren sogar durchaus hypothesenkonform.

Proband K hat die Aufgabe 1 ohne Hinweise durch den Versuchsleiter richtig gelöst. Vieles in seinem Problemlöseprozess deutet auf das Vorhandensein einer gewissen Ex-pertise hin, dies wurde auch in der retrospektiven Befragung bestätigt. Das äußerst ziel-gerichtete Vorgehen bei der Aufgabenanalyse bei gleichzeitiger Repräsentation unter Einsatz aller relevanten Wissensinhalte ist ebenfalls typisch für Experten. Aufgrund dessen würden Testpersonen wie K aber nicht zur Zielgruppe für die Hauptstudie zählen.

Die Problemlösestrategie von G war zwar nicht sehr effizient, trotzdem können bei die-ser Testperson viele allgemeine Problemlösekatégorien innerhalb des Lösungsprozesses identifiziert werden. So führte das mehrfache planvolle Aufstellen von Lösungshypothe-sen, deren Prüfung und anschließende Veränderung, die Anwendung von globalen und domänenspezifischen Heuristiken und auch von Metaheuristiken wie engagierte Herange-hensweise, Variierung der Problemlösestrategie sowie vereinfachende Umformulierung der Problemstellung zur eigenständig ausgearbeiteten korrekten Lösung. Ein Problem für G war offensichtlich, dass die anfänglich notierten Beispielzahlen zu kurz waren, um daraus eine allgemeine Regel ableiten zu können. G schrieb später zwei größere aufein-ander folgende Binärzahlen untereinander, was ihm zusammen mit der Erkenntnis, dass binärarithmetisch das Subtrahieren einer 1 von einer 0 in der selben Stelle dasselbe Er-gebnis liefert, wie das Aufaddieren einer 1, die Formulierung einer allgemeinen Regel ermöglichte.

Proband Q verfolgte einen Ansatz, bei dem jede Erkenntnis gleich in eine Implementie-rung umgesetzt wurde. Man kann hier von einem depth-first-Zugang ohne globale Über-sicht sprechen. Alle anderen Probanden gingen eher nach einem breadth-first-Ansatz vor: zuerst erfolgte die Identifizierung der einzelnen Teillösungsschritte, dann wurde der Algorithmus für die einzelnen Teillösungen allgemeingültig formuliert und danach – mit Ausnahme von G – umgesetzt. Dies verhinderte, dass einmal gefundene Teillösungen ständig an neue Erkenntnisse angepasst werden mussten. Genau das war beim Problem-löseprozess von Proband Q aber zu beobachten. Die übereilte Implementierung unge-prüfter Lösungssequenzen verhinderte letztlich das Erkennen einer allgemeingültigen Regel. Während der gesamten Problemlösephase wurde niemals vom Turingmaschinen-modell abstrahiert, vielmehr wurden alle Lösungsansätze unmittelbar auf das Modell bezogen. Der Proband zeigte keine engagierte Herangehensweise und gab sich bei dieser Aufgabe stets mit der erstbesten Lösung zufrieden. Wie für Novizen typisch, war auch bei Q das Auftreten systematischer Fehler im Problemlöseprozess erkennbar.

Beispiele für das Anwenden informatischer Problemlösekatégorien sind in den Verbal-protokollen der Probanden K und M zu finden. Bei dieser Aufgabe führte der Zugang der Probanden, ein Suchverfahren ohne konkrete Beispielzahl anzuwenden, dazu, dass die

Regel zur Dekrementierung erheblich schneller präsentiert werden konnte als durch das Vorgehen der anderen Probanden, welches sich an konkreten Zahlenfolgen orientierte und damit durch die Betrachtung von Einzelfällen ablenkte oder zu Fehlschlüssen verleitete. K und M gingen bei der Herleitung einer Dekrementierungsregel mit einem Algorithmus vor, der von hinten beginnend zunächst eine Fallunterscheidung macht. Steht an der aktuellen Stelle eine 1, wird diese in eine 0 umgewandelt und der Algorithmus terminiert. Ist dagegen die aktuelle Stelle =0, wird diese in eine 1 umgewandelt, dann der Stellenindex um 1 inkrementiert und derselbe Algorithmus mit verändertem Stellenindex auf die Ziffer der nächsten Stelle angewendet. Dieses Vorgehen kann als Anwendung der fundamentalen Idee der Parametrisierung beschrieben werden, was die Hypothese stützt, dass die Anwendung adäquater fundamentaler informatischer Prinzipien zu schnellerem Problemlöseerfolg führt. Das eben geschilderte Verfahren erinnert auch an die Art der schriftlichen Subtraktion, wenn Minuend und Subtrahend untereinander geschrieben werden und der Übertrag immer an die nächste Stelle weitergereicht wird, und hat somit auch Bezug zur Praxis.

Zusammenfassend kann gesagt werden, dass diejenigen Probanden, deren Problemlöseprozesse charakteristische allgemeine und informatische Problemlösekatogorien aufwiesen, auch die erfolgreichereren Problemlöser waren.

## Literaturverzeichnis

- [Ande81] J.R. Anderson, Cognitive Skills and Their Acquisition, Lawrence Erlbaum Associates, Publishers, 1981
- [Ande00] J.R. Anderson, Kognitive Psychologie, Spektrum Akademie Verlag, 2000
- [Borr04] R. Borromeo Ferri, Mathematische Denkstile, Franzbecker, 2004
- [Burt97] L. Burton, Mathematicians and their Epistemologies – and the Learning of Mathematics, in: Inge Schwank, (Ed.), European Research in Mathematics Education Vol. I, Osnabrück: Forschungsinstitut für Mathematikdidaktik, p. 87-102, 1999
- [Chi97] M.T.H. Chi, Quantifying Qualitative Analyses of Verbal Data: A Practical Guide, The Journal of the Learning Sciences, 6(3), 271-315, 1997
- [Deff84] G. Deffner, Lautes Denken - Untersuchungen zur Qualität eines Datenerhebungsverfahrens, Verlag Peter Lang, 1984
- [Dunc66] K. Duncker, Zur Psychologie des produktiven Denkens, Springer, 1966
- [ErSi93] K. A. Ericsson, H. A. Simon, Protocol Analysis, MIT Press, 1993
- [Frie01] G. Friege, Wissen und Problemlösen, Logos Verlag, 2001
- [Funk03] J. Funke, Problemlösendes Denken, Kohlhammer, 2003
- [GLa04] J. Gläser, G. Laudel, Experteninterviews und qualitative Inhaltsanalyse, Verlag für Sozialwissenschaften, 2004
- [GlSt98] B. G. Glaser, A.L. Strauss, Grounded Theory, Verlag Hans Huber, 1998
- [GrRe97] H. Gruber, A. Renkl, Wege zum Können, Verlag Hans Huber, 1997
- [GrZi96] H. Gruber, A. Ziegler, Expertiseforschung: theoretische und methodische Grundlagen, Westdt. Verlag, 1996
- [MaFr92] H. Mandl, H.F. Friedrich, Lern- und Denkstrategien, Hogrefe 1992
- [Mayr00] Ph. Mayring, Inhaltsanalyse, 2000
- [MüPr02] J. Müsseler, W. Prinz, Allgemeine Psychologie, Spektrum Akademie Verlag, 2002
- [NeSi72] A. Newell, H.A. Simon, Human Problem Solving, Prentice-Hall, 1972

- [ScSc04] S. Schubert, A. Schwill, Didaktik der Informatik, Spektrum Akademie Verlag, 2004  
 [Spad92] H. Spada, Allgemeine Psychologie, Hans Huber, 1992  
 [Stäu81] Th. Stäudel, Kodiersystem zur Transkription des Lauten Denkens, Memorandum des Projektes "Systemdenken", Lehrstuhl Psychologie II, Universität Bamberg, 1981  
 [StZh01] R. J. Sternberg, L.-F. Zhang, Perspectives on Thinking, Learning and Cognitive Styles, Lawrence Erlbaum Associates, Publishers, 2001  
 [WaWe90] M.R. Waldmann, F.E. Weinert, Intelligenz und Denken, Hogrefe 1990

## Anhang

### Klasse I: Allgemeine Problemlösekatogorien

Kategorie	Attribut	Dimension
<b>Problemlösephase</b>	<i>Dauer</i>	Minuten + Sekunden
<u>Subkategorien:</u> Aufgabenverständnis Repräsentation Lösungserarbeitung Evaluation	<i>Vollständigkeit</i>	vollst. ↔ unvollst.
<b>Beschreibung:</b> nach [Frie01] lässt sich der Problemlöseprozess in verschiedene Einzelphasen, wie „Aufgabenverständnis“, „Repräsentation“, „Lösungserarbeitung“ und „Evaluation“ unterteilen, die interindividuell unterschiedlich strukturiert sein können. In der Pilotphase hat sich gezeigt, dass sich diese Phasen probanden- und aufgabenabhängig sowohl in ihrer Dauer als auch in ihrer Vollständigkeit stark unterscheiden können.		

Die Kategorie „Problemlösephase“ hat die folgenden 4 Sub-Kategorien:

Kategorie	Attribut	Dimension
<b>Aufgabenverständnis</b>	<i>Qualität</i>	klar ↔ unklar
<u>Oberkategorie:</u> Problemlösephase		
<b>Beschreibung:</b> Phase zu Beginn der Problembearbeitung, in der die Aufgabenstellung unter Zuhilfenahme vorhandenen Wissens analysiert wird		

Kategorie	Attribut	Dimension
<b>Repräsentation</b>	<i>Qualität</i>	richtig ↔ falsch
<u>Oberkategorie:</u> Problemlösephase		
<b>Beschreibung:</b> Repräsentation meint die Transformation der Aufgabenstellung in eine domänenspezifische Fachsprache. Zweck ist, die Anwendung domänenspezifischer Problemlöseoperatoren zu ermöglichen.		

Kategorie	Attribut	Dimension
<b>Lösungserarbeitung</b>	<i>Richtigkeit</i>	richtig ↔ falsch
<u>Oberkategorie:</u> Problemlösephase	<i>Strukturiertheit</i>	hoch ↔ niedrig
<b>Beschreibung:</b> Eigentliche Aufgabenbearbeitungsphase, in der die Lösung erarbeitet wird.		

Kategorie	Attribut	Dimension
<b>Evaluation</b>	<i>Vollständigkeit</i>	vollständig ↔ unvollständig
<u>Oberkategorie:</u> Problemlösephase		
<b>Beschreibung:</b> Überprüfung von Lösungen oder Teillösungen. Wenn die Lösung als falsch befunden wird, setzt der Problemlöse-Prozess in einer der 3 vorherigen Phasen wieder ein.		

Weitere allgemeine Problemlösekatogorien sind u. a.:

Kategorie	Attribut	Dimension
<b>Hypothese</b>	<i>Lösungsähnlichkeit</i>	hoch ↔ gering
<u>Oberkategorie:</u> Allgemeine Problemlösestrategie		
<b>Beschreibung:</b> Hypothesen sind Vermutungen über die Lösung, die mehr oder wenig Ähnlichkeit mit der tatsächlichen Lösung aufweisen. Die Hypothesen werden in einem als Hypothesengenerierung bezeichneten Prozess entwickelt und in der (meist direkt anschließenden) Phase der Hypothesenprüfung evaluiert. Als Lösungsähnlichkeit kann die noch erforderliche Anzahl von Schritten verstanden werden, die noch bis zum Erreichen der tatsächlichen Lösung erforderlich sind und die umso niedriger ist, je ähnlicher eine Lösungsvermutung der angestrebten Lösung ist.		

Kategorie	Attribut	Dimension
<b>Hypothesengenerierung</b>	<i>Frequenz</i>	oft ↔ nie
<u>Oberkategorie:</u> Allgemeine Problemlösestrategie	<i>Vollständigkeit</i>	ja ↔ nein
	<u>Abbruchkriterium:</u> <i>rationale Falsifizierung</i> <i>fehlende Wissensinhalte</i>	
<b>Beschreibung:</b> bei der Generierung einer Hypothese wird die vorhandene Information ausgewertet und unter Zuhilfenahme von Wissensinhalten mental ein mögliches Lösungsszenario konstruiert. Dieser Prozess kann entweder vollständig ablaufen, was dann zu einer zu überprüfenden Lösungsvermutung führt oder aber durch das Erkennen einer Fehlvermutung vorher abgebrochen werden. Der Prozess der Hypothesengenerierung entfällt, wenn eine bekannte Musterlösung als Lösung vermutet wird.		

Kategorie	Attribut	Dimension
<b>Hypothesenprüfung</b>	<i>Form</i>	empirisch ↔ rational
<u>Oberkategorie:</u> Allgemeine Problemlösestrategie	<i>Genauigkeit</i>	hoch ↔ niedrig
	<i>Zielsetzung</i>	Verifikation ↔ Falsifikation
<p><b>Beschreibung:</b> im Anschluss an die Generierung einer Lösungsvermutung wird diese in der Regel überprüft. Dies kann mit einem unterschiedlichen Maß an Genauigkeit erfolgen, entweder rational, d.h. durch Überdenken oder empirisch durch Anwenden der vermuteten Lösung auf einzelne Beispiele. Fällt diese Prüfung positiv aus, wird die Hypothese zur (Teil-) Lösung erklärt. [MüPr02] beschreiben, dass sowohl von Novizen als auch Experten Hypothesen mit dem Ziel einer Verifikation der aufgestellten Hypothesen geprüft werden, d.h., es werden Evidenzen für die Richtigkeit der Hypothese gesucht, anstatt, wie es effektiver wäre, zu versuchen, die aufgestellte Hypothese zu falsifizieren.</p>		

Kategorie	Attribut	Dimension
<b>Denkstil</b>	<i>Ausprägung</i>	stark ↔ schwach
	<i>Art<sup>2</sup></i>	reflexiv intuitiv analytisch impulsiv ...
<p><b>Beschreibung:</b> der bei einer Versuchsperson vorherrschende Denkstil kann möglicherweise nur aus retrospektiven Interviews abgeleitet werden und ggf. aus der Art der verwendeten Skizzen.</p>		

Kategorie	Attribut	Dimension
<b>Heuristik</b>	<i>Frequenz</i>	oft ↔ nie
<u>Oberkategorie:</u> Allgemeine Problemlösestrategie	<i>Spezifität</i>	hoch ↔ niedrig
<u>Subkategorien:</u> Globalheuristik Domänenspezifische Heuristik Metaheuristik		
<p><b>Beschreibung:</b> Heuristiken sind wesentlicher Bestandteil des Problemlösungsprozesses. Durch sie soll der Suchraum eingeschränkt und die Problemlösung vereinfacht werden. Eine heuristische Vorgehensweise garantiert jedoch keine Lösungsfindung, diese wird aber durch die Wahl von spezifischeren Heuristiken anstelle von allgemeinen Heuristiken wahrscheinlicher. Heuristiken können nicht nur zur Lösungsfindung, sondern z.B. auch zur Repräsentation oder Evaluation eingesetzt werden.</p>		

Kategorie	Attribut	Dimension
<b>Globalheuristik</b>	<i>Frequenz</i>	oft ↔ nie
<u>Oberkategorie:</u> Heuristik	<i>Spezifität</i>	hoch ↔ niedrig
<u>Subkategorien:</u> Rekognitionsheuristik Alltagsheuristik Trial and Error Hill-climbing Mittel-Ziel-Analyse Induktion Schleifenvermeidung		
<b>Beschreibung:</b> Globalheuristiken sind unspezifisch und können daher in der Regel auf sehr viele unterschiedliche Problemklassen angewendet werden. Globalheuristiken werden meist bei unbekannteren Problemen angewendet.		

Kategorie	Attribut	Dimension
<b>Skizze</b>	<i>Frequenz</i>	oft ↔ nie
<u>Subkategorien:</u> Modellskizze Externer Speicher Zwischenrechnung	<i>Zeitpunkt</i>	in welcher Problemlöse-Phase?
	<i>Form</i>	ikonisch symbolisch
	<i>Effizienz</i>	führte die Skizze zu einer Annäherung an die Lösung? In welcher Zeit, in welchem Maß?
	<u>Zweck:</u> <i>Aufgabenverständnis Aufgabenrepräsentation Lösungserarbeitung Evaluation Hypothesengenerierung Hypothesenprüfung</i>	
<b>Beschreibung:</b> Skizzen können in verschiedenen Phasen und zu verschiedenen Zwecken angefertigt werden. Sie können sich aus ikonischen oder symbolischen Elementen zusammensetzen. Weiterhin können Skizzen zum einmaligen oder zum vernetzten Gebrauch dienen, wenn beispielsweise eine Versuchsperson zu verschiedenen Zeiten und Zielsetzungen wiederholt auf eine bestimmte Skizze zugreift. Analysiert werden soll, ob sich Einsatz, Häufigkeit, Form und Zweck der Skizzen in den einzelnen Leistungsgruppen unterscheiden.		

Kategorie	Attribut	Dimension
<b>Indikator für Vorwissen oder Bekanntheit der Aufgabe</b>  <u>Subkategorien:</u> Äußerungen der Vp Richtig verwendete (aufgaben-spezifische) Fachtermini Überspringen von ganzen Lösungsschritten	<i>Frequenz</i>  <i>Evidenz</i>	oft ↔ nie  hoch ↔ niedrig
<p><b>Beschreibung:</b> die Beantwortung der Frage nach dem Bekanntheitsgrad einer Aufgabe ist zur richtigen Interpretation der Analyseergebnisse entscheidend. Aus verschiedenen Gründen kann nicht in jedem Fall davon ausgegangen werden, dass diese Frage von der Versuchsperson zutreffend beantwortet wird. Mitunter sind also indirekte Hinweise in den Verbalprotokollen die einzige Informationsquelle.</p>		

Kategorie	Attribut	Dimension
<b>VI-Intervention</b>  <u>Subkategorien:</u> aktiv: Lösungshinweis Fehlerhinweis Aufforderung zum Lauten Denken reaktiv: Vp-Frage	<i>Frequenz</i>  <i>Zeitpunkt</i>  <u>Grund:</u> <i>fehlerhafte Aussage</i> <i>falsches Aufgabenverständnis</i> <i>völlige Ratlosigkeit, Sackgassensituation</i> <i>Frage an VI</i>  <i>Informationsgehalt</i>	oft ↔ nie  Minuten + Sekunden nach Beginn   Dimensioniert nach Klassenzugehörigkeit des Hinweises (s.o.)
<p><b>Beschreibung:</b> zunächst sei angemerkt, dass jede Intervention des Versuchsleiters den Problemlöseprozess der Versuchsperson beeinflusst. Somit müssen solche Eingriffe äußerst sparsam und nach wohldefinierten und objektiven Kriterien erfolgen. Sofort korrigiert werden sollte ein falsches Aufgabenverständnis, weil andernfalls keine vergleichenden Aussagen über den Problemlöseprozess gemacht werden können. Auch Wissensfragen können beantwortet werden, da es hier nicht um einen Wissenstest geht. Lösungs- und Fehlerhinweise sollten – das hat die Pilotstudie gezeigt – möglichst erst am Ende eines Problemlöseprozesses erfolgen, also wenn der Proband eine falsche Lösung präsentiert, wenn er nach einer längeren Phase der Ratlosigkeit erklärt, eine Aufgabe nicht lösen zu können oder wenn in einer Sackgassensituation kein weiteres Fortschreiten des Lösungsprozesses mehr zu erwarten ist. Die Hinweise sollten dann nach Möglichkeit nach dem oben beschriebenen abgestuften Kategoriensystem vermittelt werden.</p>		

Kategorie	Attribut	Dimension
<b>Umdenken</b>	<i>Frequenz</i>	oft ↔ nie
<u>Oberkategorie:</u> Allgemeine Problemlösestrategie	<i>Effektivität</i>	hoch ↔ niedrig
<b>Beschreibung:</b> Umdenken eines Problems stellt im Gegensatz zum heuristischen Vorgehen eine Erweiterung des Suchraums dar, was dann sinnvoll ist, wenn der subjektiv angenommene Suchraum kleiner als der objektive Problemraum ist und es dadurch zu keiner Lösungsfindung kommen kann.		

## Klasse II: Informatische Problemlösekatogorien

Die informatischen Problemlösekatogorien werden in Anlehnung an die fundamentalen Ideen der Informatik definiert. Geklärt werden soll die Frage, welche dieser Ideen im Lösungsprozess einer Versuchsperson zum Tragen kommt. Als Oberkategorien sind in [ScSc04] die 3 Masterideen *Algorithmisieren*, *strukturierte Zerlegung* und *Sprache* mit ihren jeweiligen Subkategorien beschrieben. Im Bereich der Algorithmisierung findet sich beispielsweise die Idee der Alternative (if, case, usw.) sowie der rekursiven Prozeduren. Die Masteridee der strukturierten Zerlegung wird u. a. durch die Idee der Hierarchisierung verfeinert, welche etwa die Zerlegung eines Objektraums in eine Baumstruktur enthält. Auch domänenspezifische Heuristiken sind Bestandteil informatischer Problemlösekatogorien.

Kategorie	Attribut	Dimension
<b>Domänenspezifische Heuristik</b>	<i>Frequenz</i>	oft ↔ nie
<u>Oberkategorie:</u> Heuristik	<i>Spezifität</i>	hoch ↔ niedrig
<u>Subkategorien:</u> First Fit – Best Fit Extremwertüberprüfung Good-suffix-Heuristik Näherungsverfahren		

## Klasse III: Aufgabenspezifische Problemlösekatogorien

Die aufgabenspezifischen Problemlösekatogorien zu Aufgabe 1 können wie folgt definiert werden:

**Schlüsselerkenntnis:** Wenn  $n > 0$ , realisiere die Funktion  $n-1$ . Wenn  $n=0$ : STOP.

**Fehlannahme:**  $n$  wird vom höchstwertigen Bit her dekrementiert.

**Teillösung 1:** Testen, ob die Zahl auf dem Turingband 0 ist. Wenn ja: STOP.  
Wenn nein:

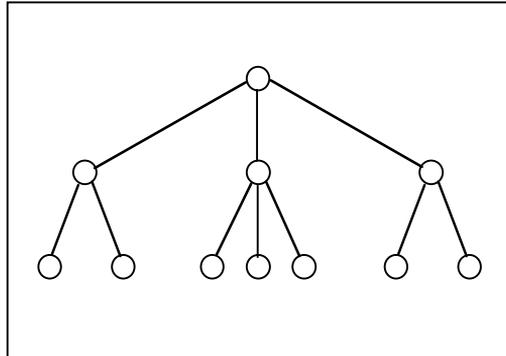
**Teillösung 2:** Bewegen des Schreib-Lese-Kopfes auf die letzte Stelle der Zahl

**Teillösung 3:** Testen, ob die aktuelle Ziffer eine 1 ist. Wenn ja, dann diese durch eine 0 ersetzen und STOP. Wenn nein:

**Teillösung 4:** Letzte Ziffer in eine 1 umwandeln, Schreib-Lese-Kopf eine Stelle nach links und mit Teillösung 3 fortfahren.

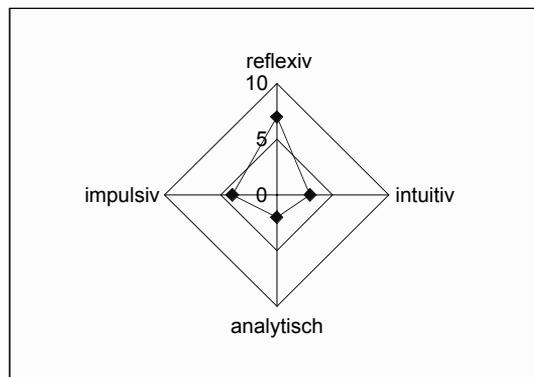
## Anmerkungen

1 Darstellung der Zerlegung in Teilprobleme als Baumdiagramm mit  $h$  (Höhe) als Maß für die Anzahl der Hierarchieebenen und  $b_0 \dots b_i$  als Maß für die Anzahl der Teilprobleme auf jeder Ebene. Untersucht werden soll, inwieweit sich die Werte für  $h$  und  $b$  bei Hochleistern und Niedrigleistern unterscheiden.



**Abbildung 2:** Baum mit  $h=2$ ,  $b_0=1$ ,  $b_1=3$  und  $b_2=7$

2 Darstellung der Ausprägung der verschiedenen Denkstile, über die eine Person verfügt, als Netzdiagramm. Interessant ist die Frage, ob die Versuchspersonen ihre Denkstile in Abhängigkeit von dem präsentierten Aufgabentyp einsetzen.



**Abbildung 3:** Netzdiagramm mit Ausprägungsmaß von 4 verschiedenen Denkstilen



# Standards für die Mittelstufeninformatik?

Eckart Modrow

Max-Planck-Gymnasium  
Theaterplatz 10  
37073 Göttingen  
emodrow@informatik.uni-goettingen.de

**Abstract:** Der Beitrag diskutiert, ausgehend von Betrachtungen zu den Aufgaben allgemeinbildender Schulen, die Auswirkungen von Standards für die Mittelstufe und die ihrer Evaluierung.

## 1 Einleitung

Allgemeinbildende Schulen haben u. a. die Aufgabe, ihre Schülerinnen und Schüler auf ein erfolgreiches Berufsleben vorzubereiten. Da dieses im internationalen Vergleich nicht effizient genug geschieht, wird das Schulwesen derzeit im Zuge der Output-Orientierung in einem bisher nicht gekannten Maße und mit erstaunlicher Geschwindigkeit umgebaut, wobei nicht nur die zu erreichenden Standards in der Form von Kompetenzformulierungen vorgegeben werden, sondern auch eine durchgehende Evaluation dieser Standards angestrebt wird. Bisher ist dieses in Kernfächern wie Mathematik, Deutsch, einer Fremdsprache und den Naturwissenschaften vorgesehen, aber auch in der Informatik gibt es Bestrebungen, in den Evaluationsprozess eingebunden zu werden. Dieser Artikel befasst sich mit der Frage, ob diese Ausweitung angesichts der weitergehenden Aufgaben allgemeinbildender Schulen eine so gute Idee ist.

## 2 Zur Rolle von „Nebenfächern“

Standards werden meist schulformübergreifend als Mindeststandards formuliert, und damit beziehen sie sich notwendigerweise auf die Mittelstufe (siehe z. B. [Pu05]). Nehmen wir als Beispiel das Gymnasium, dann befindet sich die Mittelstufe dieser Schulform für die meisten Schülerinnen und Schüler zeitlich noch recht weit vom Berufsleben entfernt, und sie hat natürlich auch noch andere Aufgaben als die effiziente Vermittlung von anwendbarem Wissen. Sie hat z. B. die Aufgabe, Schülerinnen und Schüler unabhängig von ihrem sozialen oder familiären Umfeld Kontakte zu möglichst verschiedenen Ideen, Inhalten, Methoden, Tätigkeiten und Arbeitsbereichen zu verschaffen, kurz: Orientierung zu geben. Schule dient als „Probeleben“, in dem sich Heranwachsende auf unterschiedlichen Gebieten versuchen können, wobei diese Versuche ohne gravierende Folgen auch fehlschlagen dürfen. Dabei wird gehofft, dass einige der Erprobungen er-

folgreich sind, dass die Jugendlichen mit dem Ende ihrer Schullaufbahn über eine positive Auswahl von mehreren möglichen Lebensperspektiven verfügen. Dieses Sich-Ausprobieren ist nun gerade das Gegenteil von Effizienz. Versucht die allgemeinbildende Schule, effizient zu sein, Sackgassen und Umwege zu vermeiden, dann kann sie ihre Aufgabe unter dem oben genannten Orientierungsaspekt nicht mehr erfüllen.

Bisher wurde im input-orientierten Schulwesen versucht, anhand von Inhaltskatalogen dem Bildungsauftrag gerecht zu werden. M. Thomas [Th00] formuliert das so: „*Unterricht schafft[...] eine anregende Umgebung, in der Menschen auf der Grundlage bedeutensreicher Kulturgüter in einen kulturellen Vermittlungsprozess eintreten können, in welchem sie ihre eigene wertvolle Persönlichkeit herausbilden bzw. sich zu dieser bilden können.*“ Dass das Gelernte auch ausreicht, sich später im Beruf behaupten zu können, wurde stillschweigend vorausgesetzt und kaum thematisiert. Wichtiger erschienen andere Aspekte, die sich z. B. in den bekannten Allgemeinbildungskriterien von Heymann und Bussmann [BH87] finden (*Lebensvorbereitung, Stiftung kultureller Kohärenz, Weltorientierung, Anleitung zum kritischen Vernunftgebrauch, Entfaltung von Verantwortungsbereitschaft, Einübung in Verständigung und Kooperation, Stärkung des Schülers*) oder bei Klafki [K185]: „*Die didaktisch zentrale Frage lautet dann für den Unterricht: Inwiefern sind wissenschaftliche Erkenntnisse notwendig, um diese „Lebenswelt“ durchschaubar, verstehbar und den sich entwickelnden Menschen in ihr urteilsfähig, kritikfähig, handlungsfähig werden zu lassen?*“ Dabei zeigte die Gesellschaft ein erstaunliches, m. E. berechtigtes Zutrauen zu den Unterrichtenden: denen wurden zwar die Unterrichtsinhalte vorgegeben, der zeitliche Umfang, in dem diese behandelt wurden, war aber ziemlich freigestellt und die Ergebnisse (der „Output“) wurden kaum kontrolliert.<sup>1</sup> Offensichtlich hatte die „Bildung wertvoller Persönlichkeiten“ einen höheren Stellenwert als die „Ausbildung künftiger Arbeitskräfte“ – und genau in diesem Punkt hat sich etwas geändert.

Nur in wenigen Fällen wurde bisher der Output kontrolliert – z. B. beim Zentralabitur einiger Bundesländer – und von diesen Fällen sollten wir versuchen zu lernen. Wir sollten vor allem überlegen, ob in einer Situation, in der sich die gesellschaftliche Bewertung einiger Aspekte allgemeiner Bildung geändert hat, gleich die gesamte Bildung auf eben diese Aspekte hin umgebaut werden muss. Vielleicht behalten ja auch einige der älteren Überlegungen zur Allgemeinbildung ihren Wert. Die aus dem PISA-Prozess abgeleiteten Anforderungen beziehen sich – wie ich meine, aus gutem Grund – nur auf die angegebenen Kernfächer. Wird in diesen, und damit im weit überwiegenden Teil der Unterrichtszeit, stärker als bisher der Ausbildungsaspekt betont, dann erhalten die (noch) nicht vom PISA-Prozess betroffenen Fächer, also die „Nebenfächer“, unter traditionellen Bildungsaspekten eine bedeutendere Rolle. Das gilt schon lange für Fächer wie Sport, Musik und Kunst, die auch bisher nicht so sehr wegen ihrer beruflichen Brauchbarkeit in den Schulen vertreten sind. Wir sollten es uns m. E. gut überlegen, ob wir die zukünftig noch vorhandenen Freiräume der Nebenfächer selbst einschränken.

---

<sup>1</sup> Dass das Vertrauen nicht grenzenlos war, zeigte sich z. B. an den immer weiter ausufernden Inhaltsvorgaben.

### 3 Zur Wirkung von Standards

Augenscheinlich hat die fehlende Outputkontrolle dazu geführt, dass nicht nur die Gesamtergebnisse des Bildungssystems hinsichtlich der z. B. im PISA-Prozess getesteten Aspekte zu wünschen übrig lassen, sondern dass sich die Ergebnisse zwischen den Schulen selbst zu stark unterscheiden. In dieser Hinsicht ist das Schulsystem ziemlich ungerecht geworden, weil die Ergebnisse der Leistungsmessung auch stark von der Leistungskultur der gerade besuchten Schule abhängen, und nicht nur von der individuell erbrachten Leistung. Dass hier mithilfe von Standards versucht wird, einerseits eine Verbesserung der Gesamtleistung des Systems, andererseits eine bessere Vergleichbarkeit zu erreichen, ist folgerichtig und wäre auch ohne PISA wünschenswert gewesen. Dass die Standards über Kompetenzen und Kompetenzstufen so konkret zu formulieren sind, dass sie überprüfbar werden, ist ebenfalls einsehbar, weil es sonst mit der Outputorientierung nicht weit her wäre. Interessant ist die Frage, *wie* diese Überprüfung aussieht und welche Wirkung sie hat.

Die Konkretisierung des gewünschten Outputs von Schulen hat die Konsequenz, dass genau die Kompetenzen konkretisiert werden, die sich auch evaluieren lassen – eben das ist ja das Ziel. Erfolgt die Evaluation in schriftlicher Form, was flächendeckend gar nicht anders möglich ist, dann kommen wir zu einer für ein allgemeines Bildungssystem im Wortsinn merkwürdigen Situation, dass einerseits die testbaren „harten“ Kompetenzen („Skills“) sehr viel klarer als bisher herausgestellt werden, andererseits die schriftlich nicht testbaren „weichen“ im Vagen verbleiben – und im Vergleich noch viel vager erscheinen. Diese Situation ist nicht ganz neu; sie wird aber dadurch verschärft, dass die Gewichtung zwischen diesen Bereichen sich deutlich geändert hat: nur die eine Klasse von Kompetenzen wird herangezogen, um den Output zu messen. Es wäre weltfremd anzunehmen, dass sich durch die geänderte Gewichtung zwischen harten und weichen Kompetenzen keine Verschiebungen zwischen diesen Bereichen ergeben. Es wäre auch weltfremd anzunehmen, dass die für die Evaluation wichtigen Bereiche nicht ein größeres Gewicht als bisher erhielten: sonst wäre die Outputorientierung ja wirkungslos. Damit stellt sich die Frage präziser: welche Auswirkungen ergeben sich aus der *schriftlichen* Evaluation von Kompetenzen?

Humbert und Pasternak [HP05] merken an, dass sich Qualitäten wie *Lebensvorbereitung* und *Mündigkeit* (hier: nach ISO 9000) nicht testen lassen. Wir können die Liste mühelos um Begriffe wie *Teamfähigkeit*, *Selbständigkeit*, *Persönlichkeitsentwicklung*, *soziales Verhalten*, ... erweitern. Lassen sich die nicht beurteilen? Natürlich lassen sich Kompetenzen auf diesen Gebieten ebenso formulieren wie im „harten“ Bereich<sup>2</sup>, und es gehört zu den traditionellen Aufgaben von Unterrichtenden, auch hier die Arbeit und die Lernfortschritte zu beurteilen – als *mündliche* Zensuren. Der Verzicht auf eine klare Formulierung der anzustrebenden „weichen“ Kompetenzen resultiert also allein aus dem Bedürfnis nach einer flächendeckenden Evaluierung: oder – schärfer formuliert – eine Ausrichtung auf den PISA-Prozess verhindert, dass wesentliche Teile der allgemeinbildenden

---

<sup>2</sup> Z. B. ein Versuch zur „Teamfähigkeit“: „Die Schülerinnen und Schüler teilen ein Problem in sinnvoll Teile auf, verteilen die erforderlichen Arbeiten, koordinieren diese, führen sie zusammen und testen die Ergebnisse.“

den Aufgaben der Schulinformatik hinreichend klar formuliert und eingefordert werden.

Kommen wir noch einmal auf die Erfahrungen mit dem Zentralabitur zurück. Im Vergleich zu Bundesländern ohne Zentralabitur sind Zentralabituraufgaben z. B. in Physik wesentlich gleichförmiger, standardisierter, ideenärmer, mit weniger Bezug zu echten Versuchen und erstrecken sich auf einen engeren Bereich. Sie sind schlicht langweiliger – und müssen das auch sein. Sie sind aber außerordentlich wirksam, denn das Herannahen der Abschlussprüfung bewirkt eine Fokussierung auf die möglicherweise geprüften Gebiete. Konkret: wenn Experimentieren im Zentralabitur nicht geprüft werden kann, dann wird auch weniger experimentiert. Vor diesem Hintergrund sind Aussagen, dass eine ausschließliche Outputorientierung ohne Bezug auf den Unterricht [Pu05] möglich ist, m. E. mit Vorsicht zu genießen. Dies würde ja bedeuten, dass die Standardisierung weitgehend ohne Einfluss auf die Schulen bleibt. Naheliegender wäre, dass die Art der Testfragen direkt auf den Unterricht zurückkoppelt, dass also im Unterricht das gelernt wird, was getestet werden wird. Humbert bezeichnet diesen Effekt als eine neue Art der Inputorientierung [Hu05]. Es kann wohl nicht bezweifelt werden, dass sich die Bearbeitung PISA-ähnlicher Testfragen genauso üben lässt wie anderer „Stoff“ auch<sup>3</sup>. Da weiterhin kategoriale Bildung ohne Inhalte gar nicht möglich ist, werden die getesteten Inhalte den Unterricht bestimmen. Schon die ersten Beispiele für Test-Items zeigen, dass dieser Effekt gesehen und durchaus gewollt ist: in [HP05] wird z. B. die Benutzung eines bestimmten Betriebssystems(!) mittels Testfragen „nahegelegt“.

## 4 Standards in der Mittelstufeninformatik

Fächer allgemeinbildender Schulen können nicht isoliert gesehen werden. Sie entfalten ihre Wirkung erst im Zusammenspiel mit anderen Fächern und erbringen so ihren spezifischen Beitrag zu einer allgemeinen Bildung – in der von der Gesellschaft gerade gewünschte Form. Ein in der Mittelstufe immer noch neues Fach wie die Informatik wird zu begründen haben, welche Aspekte der Allgemeinbildung in diesem Fach besser als in anderen – oder nur in diesem Fach – zu gewinnen sind. Nur mit dieser Begründung wird es das erforderliche Stundendeputat erhalten. Neben den testbaren Inhalten und Qualifikationen gehörten zu diesem Bereich bisher ziemlich unstrittig die „weichen“ Kompetenzen, die für die Projektarbeit wichtig sind, Teamfähigkeit, selbstständiges Arbeiten und Lernen. Da diese bislang durchaus nicht immer im wünschenswerten Umfang den Unterricht prägten, wäre es im Sinne einer Standardisierung dringend nötig, *gerade dafür* Kompetenzen und Kompetenzniveaus zu definieren. Im Bereich der Mittelstufeninformatik – die es flächendeckend noch gar nicht gibt – wäre es umso wichtiger, erst einmal zu ermitteln, welche Standards mit realistischen Mitteln erreichbar sind und erreicht werden sollen, und diese müssen dann präzise gefasst werden. Der m. E. vorschnelle Übergang zur Entwicklung von Testfragen, die implizit ja Standards vorgeben, verhindert damit einerseits, dass unterschiedliche Erfahrungen in der Mittelstufeninformatik gebührend analysiert, übertragen und in unterschiedlichem Umfeld überprüft wer-

---

<sup>3</sup> Ohne diese Möglichkeit wären seinerzeit die „Vorbereitungsseminare“ zum Mediziner-test wohl kaum ihr Geld wert gewesen.

den. Erst aus diesen Erfahrungen können sich dann m. E. Standards entwickeln, die in der Breite akzeptiert werden. Versuche dazu findet man z. B. unter [Mo05] oder in den Beispielen zu [Mo06]. Für noch viel bedeutsamer halte ich aber die Folge, dass die frühe Konzentration auf die Itementwicklung es verhindern kann, die im Kontext der Unterrichtsfächer wesentlichen, aber schriftlich kaum testbaren Beiträge des Informatikunterrichts zur Allgemeinbildung scharf als Kompetenzen zu formulieren und als Ziele vorzugeben – weil sie ja sowieso nicht zur Beurteilung des Unterrichtserfolgs herangezogen werden.

Es ist sinnvoll, wie von Puhlmann verschiedentlich vorgeschlagen (z. B. [Pu04]), Kompetenzen in Analogie zu den NCTM-Standards unterschiedlichen Bereichen (oder „Bändern“) zuzuordnen. Im Schema dieser Bänder können Kompetenzen dann diskutiert werden, und die Ordnung kann nach ganz unterschiedlichen Kategorien erfolgen. Eine Möglichkeit besteht darin, wie in der Mathematik Leitideen anzugeben. Für die Schulinformatik existiert ein vergleichbares Schema in Schwills fundamentalen Ideen [Sc93]. Es läge eigentlich nahe, schon aus Gründen der Vergleichbarkeit zur Mathematik, diese als Ordnungskriterium zu wählen. Es gibt aber gute Gründe für eine andere Entscheidung: Die fundamentalen Ideen der Informatik sind mitten in der Diskussion und in der derzeitigen Form noch nicht allgemein akzeptiert, beschreiben eher die Behandlung komplexer Problemstellungen und Lösungen als die in der Sek. I zu findenden einfachen Systeme und beziehen sich weitgehend auf die Oberstufeninformatik, decken den in der Sek. I stärker zu betonenden Anwendungsaspekt ungenügend ab.

Werden Kompetenzen fachlich geordnet, dann finden sich in diesen Ordnungen wenigstens teilweise die bekannten Fachgebiete der Oberstufen-Schulinformatik wieder. Wenn das so ist, dann sollte man m. E. dieses vertraute Schema ganz übernehmen, schon um die Fortschreibung für die Oberstufe zu erleichtern und den derzeit noch meist unzureichend ausgebildeten Unterrichtenden möglichst viel Bekanntes vorzusetzen. Obwohl in Zeiten der objektorientierten Modellierung (OOM) die Trennung von Daten und Algorithmen etwas antiquiert erscheint, ist festzuhalten, dass die OOM zwar aktuell ist, aber nicht das Ende der Entwicklung darstellt. Sie ist auch nicht umfassend bzgl. der Informatikinhalte. In den veröffentlichten Beispielen für die Sek. I stellt die oft zu findende Klassifizierung von Objekten allein nach den beschreibenden Daten nur eine unzureichende Annäherung an OOM dar. Deshalb wird hier an der Trennung der Bereiche „Daten“ und „Algorithmen“ festgehalten. Insgesamt kommt man dann zu einem Schema, das sich von dem Puhlmans nur unwesentlich unterscheidet.

Mithilfe eines „Bändermodells“ (s. nächste Seite) lässt sich ganz gut beschreiben, welche Gewichtung den einzelnen Bändern in den verschiedenen Altersstufen zugewiesen wird. Vor allem aber werden die Ziele deutlich, der Unterricht erhält eine Richtung. Die vom Umfang her eigentlich unsinnige Einführung eines „Theorie-Bandes“ in den Klassenstufen 5/6 wird sinnvoll, wenn so für die wenigen in dieser Altersstufe eingeführten Begriffe und Beispiele aus diesem Gebiet gezeigt wird, in welchem Kontext sie später wieder aufgenommen werden. Den früheren Altersstufen kommt so – ähnlich wie in den Naturwissenschaften – die Aufgabe zu, *informatische Begriffsbildung* anhand altersgemäßer Beispiele zu betreiben. Da noch in absehbarer Zeit ein wesentlicher Teil des informatischen Mittelstufenunterrichts entweder in fremde Fächer integriert oder fach-

fremd von nicht dafür ausgebildeten Lehrerinnen und Lehrern unterrichtet werden wird, erhält diese Orientierung eine gesteigerte Bedeutung. Den Unterrichtenden muss gesagt werden, welche Aspekte ihres Unterrichts „richtigen“ Informatikunterricht vorbereiten, wie ihr Unterricht zu akzentuieren ist. In [Mo04] z. B. wird versucht, die in Schulen übliche Einführung in die Nutzung von Präsentationssystemen so zu gestalten, dass sie für den Informatikunterricht fruchtbar wird. Auch die Gestaltung der Mittelstufeninformatik in Bayern (z. B. in [Hu00]) als Gegensatz zu einer Benutzerschulung kann so verstanden werden.

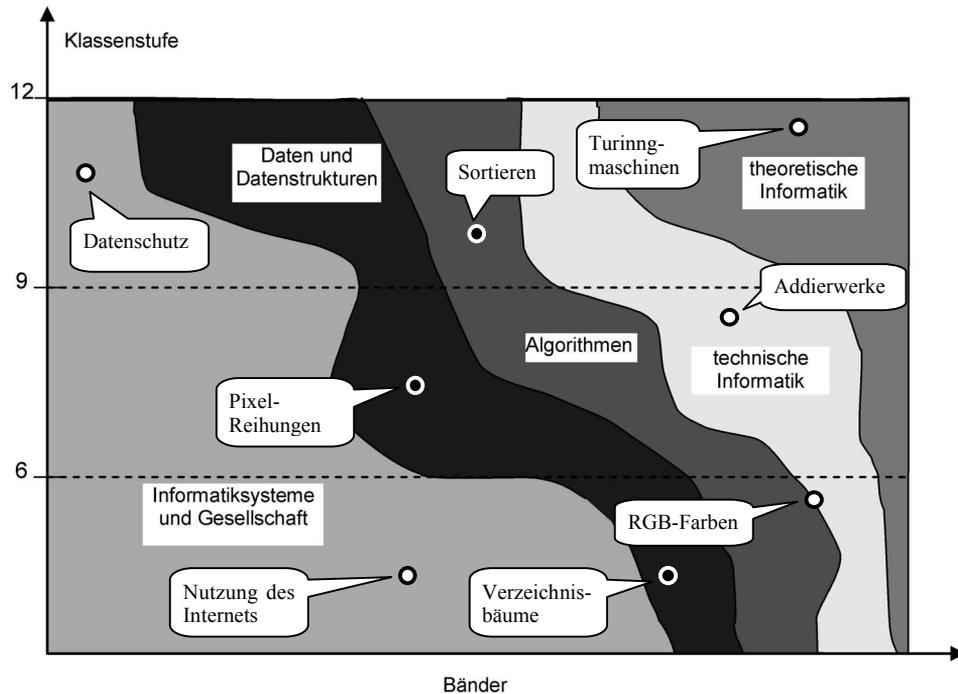


Abbildung 1: Bänder als Ordnungsschema für Kompetenzen  
An einzelnen Punkten wurden beispielhaft Unterrichtsthemen markiert.

## 5 Fazit

Eine Standardisierung scheint mir im Bereich der Mittelstufeninformatik dringend erforderlich, sie scheint mir aber ohne zureichende Erfahrung und Diskussion dieser Erfahrung kaum machbar, denn ein neues Fach entsteht nun mal nicht am Schreibtisch. Es muss Beispiele für guten Unterricht geben, der die Standards erreicht. Deshalb sollten Standards aus der Schnittmenge guten Unterrichts folgen. Da die Beiträge der Mittelstufeninformatik m. E. nicht nur bei den Inhalten und Methoden, sondern besonders im Bereich der sozialen und der persönlichen Entwicklung liegen, scheint es mir angemessen, das Fach im angegebenen Sinn als ein „Nebenfach“ zu behandeln, das es – hoffentlich – einmal flächendeckend sein wird. Wenn sich die am PISA-Prozess beteiligten

Kernfächer auf die Ausbildungsaspekte der Bildung konzentrieren müssen, dann gewinnt ein so facettenreiches Fach wie die Schulinformatik seine Bedeutung gerade aus der Betonung der nicht testbaren Gebiete. Wir sollten dem Fach Zeit geben, sich in dieser Richtung zu entwickeln.

## Literaturverzeichnis

- [BH87] Bussmann, H.; Heymann, H.-W.: Computer und Allgemeinbildung. Neue Sammlung 1, 1987
- [Hu05] Humbert, L.: Didaktik der Informatik. Teubner 2005
- [HP05] Humbert, L.; Pasternak, A.: Informatikkompetenzen. In LOG IN 135, 2005
- [Hu00] Hubwieser, P.: Didaktik der Informatik. Springer, 2000
- [Kl85] Klafki, W.: Neue Studien zur Bildungstheorie und Didaktik, Beltz, 1985
- [Mo04] Modrow, E.: Einführung in die Algorithmik. In „Schulversuch InTech - Informatikunterricht mit technischen Aspekten“. <http://www.vlin.de/sek1.html>
- [Mo05] Modrow, E.: Technik im Informatikunterricht der Sek. I. siehe [Mo04]
- [Mo06] Modrow, E.: Zur Ordnungswirkung fundamentaler Ideen der Informatik am Beispiel der theoretischen Schulinformatik. *informatica didactica* 6, 2006, <http://www.informatikdidaktik.de/InformaticaDidactica/Modrow2006>
- [Pu04] Puhlmann, H.: in Magenheimer, Schubert: concepts of empirical research and standardisation of measurement in the area of didactics of informatics. GI-Dagstuhlseminar GI-Edition, Lecture Notes in Informatics, 2004
- [Pu05] Puhlmann, H.: Standards für die Schulinformatik. In LOG IN 135, 2005
- [Sc93] Schwill, A.: Fundamentale Ideen in Mathematik und Informatik. Script, 1993
- [Th00] Thomas, M.: Einführung in die Didaktik der Informatik. Begleitmaterial zur Vorlesung, Potsdam, 2000/01



# Begriffliche Strukturen der Informatik: Ein empirischer Zugang

Markus Schneider

Didaktik der Informatik – Technische Universität München  
Boltzmannstr. 3  
85748 Garching  
markus.schneider@in.tum.de

**Abstract:** Die begrifflichen Strukturen der Informatik sind nicht nur von rein akademischen Interesse, sondern sie bilden auch die theoretische Grundlage der Entwicklung von Curricula an Schule und Hochschule. Da ist zum einen die in den letzten Jahren in den Blickpunkt didaktischer Arbeit gerückte Frage nach Standards für das Fach Informatik und zum anderen die Konzeption von Studienabläufen im Rahmen des neu eingeführten Bachelor/Master-Systems. In der vorliegenden Arbeit sollen insbesondere solche begrifflichen Strukturen identifiziert werden, die vom informatischen Anfangsunterricht an der Schule bis zu Einführungsvorlesungen an der Hochschule auftreten. In einer konzeptionellen Analyse von Übungsaufgaben an Schule und Hochschule werden die darin enthaltenen fundamentalen Ideen der Informatik identifiziert. Dadurch lassen sich Ideen identifizieren, die sich durch die gesamte Lehre der Informatik ziehen. Die Arbeit schließt mit einer kritischen Reflexion zu Details der fundamentalen Ideen.

## 1 Begriffliche Strukturierung: Wozu?

Welchen Nutzen hat eine wissenschaftliche Disziplin und nicht zuletzt auch deren Fachdidaktik von der Kenntnis der jeweiligen inhaltlichen Strukturen? Die begriffliche Strukturierung ist für jede Wissenschaft zum einen aus rein fachsystematischen Gründen von entscheidender Bedeutung. Insbesondere die Informatik, die mitunter als Strukturwissenschaft bezeichnet wird [We71], darf ihre eigene Strukturierung keinesfalls zurückstellen.

Darüber hinaus gibt es für einen in der Didaktik der Informatik Tätigen jedoch auch praktischere Gründe, sich mit der inhaltlichen Struktur seines Faches zu beschäftigen: Aufgaben wie die Gestaltung und der Entwurf von Vorlesungen bzw. Seminaren, oder die Konzipierung von Unterrichtsabläufen sind Herausforderungen, mit denen ein Lehrender tagtäglich konfrontiert ist. Zwar wird dem Lehrenden an der Schule die Grobplanung durch Lehrpläne erleichtert, jedoch bleibt die Feinstrukturierung letztlich jedem selbst überlassen. Für diese Arbeit benötigt jeder Lehrer eine begriffliche Strukturierung, die die fachlichen und didaktischen Abhängigkeiten auf unterschiedlichem Komplexitätsniveau, von der Grundschule bis hin zur Lehre an der Hochschule dokumentiert.

## 2 Vorhandene Klassifikationen

Selbstverständlich existieren bereits Klassifikationen der wichtigsten Begriffe der Informatik. Einige zentrale sollen hier angesprochen und bewertet werden. Dabei wird zwischen Klassifikationen unterschieden, die sich primär an den verschiedenen Spezialdisziplinen der Informatik orientieren und denen, die übergeordnete Prinzipien als entscheidendes Kriterium verwenden. Entsprechend werden erstere im weiteren als domänenorientierte und letztere als prinzipienorientierte Klassifikationen bezeichnet.

### 2.1 Domänenorientierte Klassifikationen

**ACM-Klassifikation:** Diese Klassifikation ([Ne98]) ist eine Taxonomie für Begriffe der Informatik, die mit der Version aus dem Jahre 1998 einen vorläufigen Endzustand erreicht hat. Sie wurde bereits im Jahre 1964 in einer ersten Version veröffentlicht, woraus eine entsprechend umfangreiche und nahezu alle Bereiche der Informatik umfassende Klassifikation resultierte. Die Intention zur Erstellung dieses Klassifikationsschemas war jedoch keine didaktische, sondern eher eine bibliographische. In der Version von 1982 nennt sich die Klassifikation noch „Computing Reviews Classification System“, was darauf hindeutet, das man in erster Linie ein System zur Klassifikation der in der Informatik auftretenden wissenschaftlichen Literatur im Auge hatte. Folglich wird die oberste Ebene der Taxonomie durch Begriffe repräsentiert, die stellvertretend für die verschiedenen Spezialdisziplinen der Informatik stehen: „Hardware“, „Software“, „Data“, „Theory of Computing“ um nur einige zu nennen. Leider lassen sich dann viele übergeordnete Begriffe, wie beispielsweise die Rekursion, nur schwer in die Klassifikation einfügen: Sie findet sich auf einer relativ tiefen Ebene des Klassifikationsbaumes bei den Programmierkonstrukten, obwohl Rekursion sicherlich bei zahlreichen anderen Begriffen der Informatik, wie etwa bei Design-Pattern, formalen Sprachen oder rekursiven Datenstrukturen eine Rolle spielt (vgl. z.B. [Sc02]).

**Domänenorientierte Klassifikation von P. Denning [De99]:** Eine weitere domänenorientierte Klassifikation wurde von P. Denning vorgeschlagen [De99]. Es handelt sich hier um eine zweidimensionale Klassifikation: Fachlich-inhaltliche Kategorien, wie Algorithmen und Datenstrukturen, Programmiersprachen, Netzwerke, AI, Datenbanken bilden eine Dimension. Orthogonal dazu steht eine Kategorisierung in unterschiedliche Abstraktionsniveau: Theorie, Abstraktion, Design. Im Falle der inhaltlichen Kategorie „Programmiersprachen“ sind Konzepte wie  $\lambda$ -Kalkül oder Turing-Maschinen der Theorie, die verschiedenen Programmierparadigmen (objektorientiert, funktional, etc.) der Abstraktion und die konkreten Programmiersprachen der Designebene zugeordnet.

**Resümee:** Das entscheidende Defizit domänenorientierter Klassifikationen ist offensichtlich: Durch Aufspaltung der Inhalte in die verschiedenen Teildisziplinen fällt es dem Lernenden schwer, Konzepte, die in allen Teilbereichen auftreten, als solche zu erkennen. Lehrstrategien, die ausschließlich auf derartigen Klassifikationen aufbauen, bergen somit die Gefahr, isoliertes und punktuelles Wissen aufzubauen, anstatt die Begriffe zu vernetzen und übergeordnete Strukturen zu bilden.

## 2.2 Prinzipienorientierte Klassifikationen

Eine vollkommen andere Strukturierungsstrategie weisen prinzipienorientierte Klassifikationen, wie die so genannten fundamentalen Ideen von A. Schwill [Sc93] oder die „Great Principles“ von P. Denning [De03], [De06] auf.

**Fundamentale Ideen:** Ohne auf die Details der fundamentalen Ideen einzugehen, seien hier die entscheidenden Elemente vorgestellt, da diese in dieser Arbeit von zentraler Bedeutung sind: Der Softwareentwicklungsprozess als eine der zentralen Aufgaben der Informatik bildet den Ausgangspunkt der „Ideenfindung“. Diesem werden die so genannten „Masterideen“ Algorithmisierung, Strukturierte Zerlegung und Sprache zugeordnet. Die Masterideen wiederum enthalten verschiedene Ideengruppen wie etwa „Entwurfparadigmen“ oder „Ablauf“. Die einzelnen Ideengruppen werden schließlich mit den eigentlichen Ideen „aufgefüllt“. So enthält etwa die Ideengruppe der Programmierkonzepte die Ideen: Konkatenation, Alternative, Iteration, Rekursion, Parametrisierung und Nichtdeterminismus. Innerhalb der Ideengruppe „Hierarchisierung“ werden zwei weitere Untergruppen namens „Darstellung“ und „Realisierung“ eingeführt. Diese enthalten die Ideen der Schachtelung, des Baumes, der Klammerung und der Einrückung bzw. Compilierung und Interpretation. Die Masteridee der „Sprache“, die ursprünglich lediglich die Ideengruppen „Syntax“ und „Semantik“ beinhaltete, wurde von E. Modrow weiterentwickelt [Mo06].

**Great Principles:** Einen völlig anderen Weg als in der Arbeit aus dem Jahre 1999 [De99] beschreibt P. Denning in den Publikationen über „Great Principles of Computing“ [De03], [De06]. Zentrales Element der Klassifikation sind die so genannten „Great Principles“, die sich in verschiedenen Teilgebieten der Informatik und innerhalb eines Gebietes an zahlreichen Stellen wieder finden lassen [De06]; so sind etwa die P–NP Problematik, der Algorithmusbegriff, oder die verschiedenen Zeitkomplexitäten zu den Great Principles zu zählen. Dabei verwendet er für ein derartiges „Great Principle“ eine Definition, die der Definition der fundamentalen Idee, wie sie von Bruner [Br60] gegeben wird, verblüffend ähnlich ist, wie wohl Denning sich nicht darauf bezieht. Demzufolge ist die Liste der „Great Principles“ mit den fundamentalen Ideen von Schwill durchaus vergleichbar, wenn gleich sie nicht identisch ist. Allerdings scheint die endgültige Formulierung aller Prinzipien noch nicht abgeschlossen zu sein, denn Denning hat diese bisher nur elektronisch als vorläufige Liste veröffentlicht [De06]. Des weiteren ist die Anzahl der Prinzipien im Vergleich zum Schwillschen Ideenkatalog sehr umfangreich und eine detaillierte Beschreibung, wie die Auswahl der Prinzipien erfolgt, nicht verfügbar.

## 2.3 Folgerung

Wegen der oben angesprochenen Problematik mit domänenorientierten Klassifikationen, verbleiben aus didaktischer Sicht als Arbeitsgrundlage für eine Klassifikation der zentralen Begriffe der Informatik in erster Linie die prinzipienorientierten Klassifikationen. Da jedoch der Ansatz von Denning und insbesondere dessen Ideenkatalog noch nicht abgeschlossen ist, konzentriert sich diese Arbeit im Weiteren auf die Schwillschen Ideen. Der Schwillsche Ansatz, wie auch dessen Modifikationen nach Modrow und nicht zuletzt die „Great Principles“ von Denning werfen jedoch eine entscheidende Frage auf: Handelt es

sich bei den fundamentalen Ideen empirisch nachweisbar um übergeordnete Prinzipien der Informatik? Erfüllen Sie tatsächlich die Brunersche Forderung nach Fundamentalität [Br60]? Kritische Aufsätze zu den fundamentalen Ideen gibt es durchaus: Etwa von R. Baumann [Ba98], der nicht nur Teile des Katalogs der Ideen, sondern auch deren hierarchische Anordnung ablehnt. Jedoch wurde bisher noch nicht systematisch und empirisch untersucht, welche der verschiedenen Ideen den Brunerschen Kriterien genügen. Hierzu will diese Arbeit durch die detaillierte Untersuchung der Masterideen „Strukturierte Zerlegung“ und „Algorithmisierung“ einen Beitrag leisten. Insbesondere werden die Weiterentwicklungen von Modrow aus Platzgründen nicht diskutiert.

## 3 Fundamentale Ideen in Schule und Hochschule

### 3.1 Fundamentale Ideen in einführenden Vorlesungen zur Informatik

Während die fundamentalen Ideen ursprünglich für eine Verwendung im Schulbereich konzipiert wurden, soll hier auch untersucht werden, ob sie auf inhaltliche Strukturen der Hochschuldidaktik anwendbar sind. Betrachtet werden hierzu die Vorlesungen zur Einführung in die Informatik I aus den Jahren 2001 bis 2004 an der TU München [Br01, Bru02, Kn03, Se04]. Dabei wird analysiert, inwieweit die Lernziele von Vorlesungen im Grundstudium der Informatik durch die fundamentalen Ideen repräsentiert werden. Unter der Annahme, dass die Übungsaufgaben der Vorlesungen die Lernziele der Vorlesung repräsentieren, werden diese Aufgaben auf ihre Beziehung zu den fundamentalen Ideen analysiert.

Folgendes Beispiel einer Übungsaufgabe zur Vorlesung [Bru03] veranschaulicht die prinzipielle Vorgehensweise, wie einer Übungsaufgabe die entsprechende fundamentale Idee zugeordnet wird:

**Übungsaufgabe:** Ein Buch ist charakterisiert durch den Namen des Autors, den Buchtitel und die Verlagsdaten. Die Verlagsdaten umfassen den Namen des Verlags, die Nummer der Auflage und das Erscheinungsjahr.

a) Stellen Sie die Klasse Buch in UML dar und definieren Sie sie in Java. Geben Sie dabei auch eine Konstruktordefinition an.

b) Es soll ein Objekt der Klasse Buch erzeugt werden, das das für die Vorlesung als weiterführende Literatur empfohlene Buch von Professor Brügge repräsentiert. B. Brügge und A. Dutoit: *Object-Oriented Software-Engineering: Using UML, Patterns and Java* 2. Auflage, Prentice Hall (2003)

Stellen Sie das Objekt in UML dar und instanziiieren Sie es in Java. Das Objekt soll mit dem Namen `brueggeObjektorientierung` bezeichnet werden.

Da die fundamentalen Ideen nicht direkt die Begriffe der objektorientierten Modellierung enthalten, kann die Zuordnung nicht durch Extrahieren bestimmter Worte erfolgen. Vielmehr müssen die einzelnen Fragestellungen der Aufgabe herausgearbeitet und einer fundamentalen Idee zugeordnet werden. Obige Aufgabe ergibt folgende Zuordnung von Teilproblemen und fundamentalen Ideen:

- Aus der angegebenen Spezifikation müssen sinnvolle Klassen abgeleitet werden → Top-Down, Black-Box
- Den Klassen müssen sinnvolle Attribute zugeordnet werden → Parametrisierung
- Die Klassen sind als neue Datentypen zu interpretieren → Abstrakter Datentyp
- Es sind konkrete Objekte zu instanziiieren → Lokalität von Objekten
- Es ist eine Implementierung in Java durchzuführen → Übersetzung

In derartiger Weise wurden nun alle Übungsaufgaben der Vorlesungen [Br01, Bru02, Kn03, Se04] analysiert. Die verschiedenen Ideen wurden hierbei nur tabellarisch aufgeführt, d.h. die hierarchische Ordnung blieb unberücksichtigt. Die Tabellen 1 und 2 zeigen die Ergebnisse dieser Analyse. Die verschiedenen Einträge geben hierbei wieder, wie oft eine bestimmte Idee in den Übungsaufgaben der verschiedenen Vorlesungen thematisiert wurde. Aus Gründen der Übersicht sind hierbei Ideen, die der Masteridee „Strukturierte Zerlegung“ und solche, die der Masteridee „Algorithmisierung“ zugeordnet sind, in unterschiedlichen Tabellen aufgeführt.

### 3.2 Masteridee „Strukturierte Zerlegung“

	2001/2002	2002/2003	2003/2004	2004/2005
Top-down	0	7	> 10	> 10
Bottom-up	5	2	4	1
Black-box	> 10	> 10	> 10	> 10
Lokalität von Objekten	1	0	5	7
Spezifikation	4	5	2	4
Abstrakter Datentyp	> 10	> 10	> 10	> 10
Teamarbeit	> 0	> 0	> 0	> 0
Schachtelung	> 10	5	0	2
Baum	1	4	4	4
Klammerung	> 10	4	6	0
Einrückung	0	0	0	0
Übersetzung	0	0	> 10	> 10
Interpretation	> 10	> 10	> 10	0
Operationale Erw.	0	0	0	0
Emulation	1	0	0	0

Tabelle 1: Ideengruppe „Strukturierte Zerlegung“ in Einführungsvorlesungen zur Informatik

Tabelle 1 zeigt, dass über die Jahre hinweg nahezu jede Idee der Gruppe „Strukturierte Zerlegung“ thematisiert wird. Einzig die Idee der Operationalen Erweiterung wird von keiner der Vorlesungen tangiert. Zwar trifft dies auch auf die Einrückung zu, jedoch handelt es sich bei der Einrückung um eine Technik, die unausgesprochen bei vielen Implementierungen eingesetzt wird, wenn sie auch zur Lösung nicht notwendig ist. Darüber hinaus dürfte es sich bei Klammerung, Schachtelung und Einrückung um verwandte Begriffe zur Repräsentation von Baumstrukturen handeln. Insofern könnte man diese

drei Ideen auch als spezielle Repräsentationen von Bäumen ansehen und mit diesen zu einer Idee zusammenfassen.

Darüber hinaus fällt auf, dass manche Ideen in der einen Vorlesung häufig, in der anderen dagegen überhaupt nicht auftreten, beispielsweise die Idee des „Top-down-Entwurfs“. Diese Idee tritt in der Vorlesung [Br01] nicht, in der Vorlesung [Bru03] dagegen sehr häufig auf. Grund hierfür ist das Design der Vorlesung. Die Vorlesung [Br01] startet mit Boolescher Algebra, thematisiert anschließend ausführlich die Elemente des funktionalen Paradigmas und gelangt schließlich zu imperativen Strukturen. Eine derartige Strategie legt auch für die zugehörigen Übungen eine Vorgehensweise von kleinen Funktionen zu großen Systemen und damit den Bottom-Up-Entwurf nahe. Andererseits wurde in [Bru03] ein systemorientierter Ansatz gewählt; objektorientierte Modellierung bestimmte von Beginn an die Vorlesung. Konsequenterweise wurden auch die entsprechenden Übungsaufgaben in diesem Sinne konzipiert, die Top-Down-Strategie ist deshalb das bestimmende Entwurfsschema. Die oben angegebene Übungsaufgabe zur Modellierung und Implementierung eines Buches ist ein Beispiel hierfür.

Eine Bemerkung sollte noch zu der Idee der Emulation, die der Ideengruppe der Orthogonalisierung zu zurechnen ist, gemacht werden. Diese Idee wird in den 4 Vorlesungen nur ein einziges Mal angesprochen und auch da, genau genommen, nur am Rande. In allen anderen Vorlesungen werden Probleme aus dem Bereich der Orthogonalisierung dagegen nicht thematisiert

### 3.3 Masteridee „Algorithmisierung“

Wenden wir uns nun der Umsetzung von Ideen aus dem Bereich der Algorithmisierung zu (Tabelle 2). Die Ideen, die die klassischen Kontrollstrukturen repräsentieren (Alternative, Konkatenation, Rekursion bzw. Iteration) sowie die Ideen der Parametrisierung werden sehr häufig thematisiert. Andererseits werden die Ideen, die der Gruppe der Entwurfparadigmen und des Ablaufs zu zuordnen sind nur am Rande oder gar nicht angesprochen. Verifikations- und Komplexitätsaspekte werden ebenfalls nicht in allen Vorlesungen diskutiert.

	2001/2002	2002/2003	2003/2004	2004/2005
Branch & Bound	0	0	0	0
Divide & Conquer	1	0	2	1
Greedy	0	0	0	0
Line-Sweeping	0	0	0	0
Backtracking	0	3	0	1
Konkatenation	> 10	>10	>10	>10
Alternative	> 10	>10	>10	>10
Iteration	5	2	> 10	> 10
Rekursion	>10	> 10	> 10	> 10

Parametrisierung	> 10	> 10	> 10	> 10
Nichtdeterminismus	1	1	2	0
Nebenläufigkeit	0	0	0	3
Prozess	0	0	0	0
Prozessor	0	0	0	0
Part. Korrektheit	6	1	3	0
Terminierung	3	1	2	0
Konsistenz	0	0	0	0
Vollständigkeit	0	0	0	0
Fairness	0	0	0	0
Reduktion	0	0	0	0
Diagonalisierung	0	0	0	0
Ordnung	1	1	3	0
Unit-/ Log - cost	2	0	0	0
Worst/average case	1	0	0	0

Tabelle 2: Ideengruppe „Algorithmisierung“ in Einführungsvorlesungen der Informatik

#### 4 Fundamentale Ideen im Informatikunterricht

Im Bereich der Schulcurricula ist eine Einschränkung erforderlich: Exemplarisch wird der Lehrplan der 6. und 7. Jahrgangsstufe des Pflichtfaches Natur und Technik an bayerischen Gymnasien, in dem zu einem Drittel der Unterrichtszeit Informatik gelehrt wird, herangezogen. Auch hier sind die Übungsaufgaben Grundlage der Ideenanalyse. Konkret werden die Aufgaben der offiziell genehmigten Unterrichtswerke „Informatik I“ [Fr04] und „Ikarus: Natur und Technik“ [Br04] herangezogen. Zur Erläuterung der prinzipiellen Vorgehensweise auch hier eine Beispielaufgabe (Ikarus, II/1 Aufgabe 4):

**Attribute von Ländern:** Vergleiche in einer Tabelle die Länder Deutschland, Österreich, USA und China bezüglich der Attribute Kontinent, Größe, Einwohnerzahl, Binnenland und Hauptstadt. Schlage dazu in einem Lexikon oder Atlas nach. Wähle ein weiteres Land deiner Wahl und trage es mit allen Werten in die Tabelle ein.

Diese Übungsaufgabe ist hinsichtlich der Begriffe durchaus mit der in Abschnitt 3 erwähnten Modellierungsaufgabe vergleichbar. Auch hier müssen sinnvolle Objekte definiert und Attribute festgelegt werden. Top-Down-Zerlegung, Black-Box-Denken, Parametrisierung und Lokalität von Objekten tritt also auch hier auf, wenngleich diese Begriffe nicht explizit thematisiert werden. Andererseits wird der Klassenbegriff noch nicht tangiert, weshalb hier die Idee des „Abstrakten Datentyps“ noch keine Rolle spielt.

Das Resultat der Analyse der Übungsaufgaben der genannten Schullehrbücher zeigt Tabelle 3.

	Ikarus	Informatik I
Top-down	> 20	> 20
Bottom-up	0	0
Black-box	> 20	> 20
Lokalität von Objekten	> 20	> 20
Spezifikation	> 0	> 0
Abstrakter Datentyp	> 20	> 20
Teamarbeit	> 0	> 0
Schachtelung	> 10	> 10
Baum	> 10	> 10
Klammerung	0	0
Einrückung	0	0
Übersetzung	0	0
Interpretation	> 10	> 10
Operationale Erweiterung	0	0
Orthogonal./Emulation	0	0

Tabelle 3: Strukturierte Zerlegung in Jahrgangstufe 6/7 an bayerischen Gymnasien in Natur und Technik / Informatik

Offensichtlich gibt es auch hier Ideen, die in keiner Aufgabe thematisiert werden. Dazu gehören die Idee des Bottom-Up-Entwurfs, die Ideen der Klammerung, der Einrückung und der Übersetzung, sowie die Ideen der Operationalen Erweiterung und der Orthogonalisierung. Die Idee „Teamarbeit“ wurde mit „>0“ bewertet, da implizit bei vielen Aufgaben Gruppenarbeit erwartet wird. Ähnlich wurde Spezifikation behandelt, wenn Methoden informell beschrieben, jedoch nur eine Modellierung des Methodenkopfes verlangt ist.

Auf die Angabe der Häufigkeitstabelle für die Ideengruppe der Algorithmisierung wurde hier aus Platzgründen verzichtet. Es ergibt sich eine Tabelle, die im Vergleich zu Tabelle 2 stark „ausgedünnt“ ist: Es werden lediglich in der Ideengruppe der Programmierkonzepte die Ideen „Konkatenation“, „Alternative“ und „Iteration“ konkretisiert. Alle anderen Ideen werden nicht angesprochen.

## 5 Folgerungen

Welche Konsequenzen lassen sich aus den Überlegungen von Abschnitt 3 und 4 ziehen? Offensichtlich gibt es fundamentale Ideen, die von Beginn des Informatikunterrichts bis zur Hochschule wiederholt in unterschiedlichem Kontext diskutiert werden. Im Falle der Masteridee „Algorithmisierung“ sind es die Ideen Konkatenation, Alternative und Itera-

tion, bei der Masteridee „Strukturierte Zerlegung“ die Ideen Top-down/Bottom-Up-Methode, Geheimnisprinzip, Lokalität von Objekten, Spezifikation, Abstrakter Datentyp, Teamarbeit, Schachtelung, Klammerung und Baum (wobei letztere zu einer Idee zusammengefasst werden könnten). Diese Ideen werden auch in den weiteren Jahrgangsstufen des Bayerischen Gymnasiums im Schulfach Informatik thematisiert. Für die Jahrgangsstufen 9 und 10 wird dies vom Autor dieser Arbeit in [Sc06] gezeigt. Diese Ideen bilden somit rote Fäden, die sich durch die gesamte Lehre in der Informatik ziehen, und genügen dem von Bruner [Br60] für Ideen geforderten vertikalen Kriterium. Allerdings gibt es auch Ideen, die in den ersten Jahren des Faches Informatik (Jahrgangstufe 6/7) an bayerischen Gymnasien nur implizit oder gar nicht tangiert werden. Dazu gehören etwa die gesamten Ideen zur Evaluation, Orthogonalisierung, die meisten Entwurfsparadigmen und Ablaufaspekte.

Problematisch gestaltete sich die Zuordnung des Themenbereichs „Objektmodellierung durch Klassen- und Objektdiagramme“. Grundsätzlich müsste er der „Strukturierte Zerlegung“ zuzuordnen sein. Bei der Vernetzung der Klassen ergeben sich jedoch im Allgemeinen nichthierarchische Strukturen, weshalb derartige Graphen nicht der Ideen-Gruppe „Hierarchisierung“ zuzuordnen sind, während der Baum als spezieller Graph sich in dieser Gruppe befindet! Sollten vielleicht auch Ideen für allgemeine Graphen in die Gruppe „Hierarchisierung“ aufgenommen werden und die Benennung dieser Gruppe verallgemeinert werden?

Schließlich gibt es wichtige Konzepte, die nicht oder nur schwer im Baum der Ideen unterzubringen sind: Dazu gehört zunächst der Problembereich „Information und Repräsentation“: Bereits die erste Lehreinheit des bayerischen Curriculum für Informatik aber auch die oben diskutierte Einführungsvorlesung [Bru03] widmen sich dieser Problematik. Auch hierbei handelt es sich um einen roten Faden, der in einer begrifflichen Strukturierung des Faches repräsentiert sein müsste.

## Literaturverzeichnis

- [Ba98] Baumann, R.: Fundamentale Ideen der Informatik – gibt es das?; in „Informatische Bildung in Deutschland, Perspektiven für das 21. Jahrhundert“, Koerber, B. und Peters I.
- [Br01] Broy, M.: Einführung in die Informatik I, WS 2001/2002  
<http://www4.in.tum.de/lehre/vorlesungen/info1ws01/tutoruebung.html> (29.7.2006)
- [Br04] Brichzin, P. et al.: Ikarus, Natur und Technik 6/7, Oldenbourg Schulbuchverlag 2004
- [Br60] Bruner, J.S.: The process of education, Cambridge Mass. 1960 (dt. Übers.: "Der Prozeß der Erziehung", Berlin 1970)
- [Bru03] Brügge, B.: Einführung in die Informatik I, WS 2003/2004  
<http://www.bruegge.in.tum.de/twiki/bin/view/Lehrstuhl/InformatikI/TutorWiSe2003> (29.7.2006)
- [De99] Denning, P.: Computer Science: The Discipline, Encyclopedia of Computer Science, A. Ralston and D. Hemmendinger, eds., Nature Publishing Group, 2000, pp. 405-419.
- [De03] Denning, P.: Great Principles of Computing, Comm. ACM, Nov. 2003, pp. 15-20.
- [De06] Denning, P.: A Preliminary Listing of Great Principles,  
[http://cs.gmu.edu/cne/pjd/GP/gp\\_list.html](http://cs.gmu.edu/cne/pjd/GP/gp_list.html) (29.7.2006)
- [Fr04] Frey, E., Hubwieser, P., Winhardt, F.: Informatik I: Objekte, Strukturen, Algorithmen, Ernst Klett Verlag 2004
- [Kn03] Knoll, A.: Einführung in die Informatik I, WS 2002/2003 <http://atknoll1.informatik.tu->

- muenchen.de:8080/tum6/lectures/courses/ws0203/infot1 (27.7.2006)
- [Mo06] Modrow, E.: Zur Ordnungswirkung fundamentaler Ideen der Informatik am Beispiel der theoretischen Schulinformatik, *Informatica Didactica* 6 (2006)
- [Ne98] Coulter, N.: ACM Computing Classification System, <http://www.acm.org/class/1998/> (15.3.2006)
- [Se04] Seidl, H.: Einführung in die Informatik I, WS 2004/2005; <http://www.seidl.informatik.tu-muenchen.de/> (29.7.2006)
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. *Zeitschrift für Didaktik der Mathematik*, 1993/1
- [Sc02] Schneider, M.: Rekursive Strukturen in Einführungsvorlesungen der Informatik; (2002) *Lecture Notes in Informatics, Volume 22*, pp. 77
- [Sc06] Schneider, M.: Functional modelling, fundamental ideas and threads in the subject informatics, ISSEP 2006, Vilnius (Litauen)
- [We71] Weizsäcker von, C.F.: *Die Einheit der Natur*, 1971 Deutscher Taschenbuch Verlag

# Unterrichtsmodellentwicklung zur Förderung des Informatiksystemverständnisses mit Entwurfsmustern

Peer Stechert

Didaktik der Informatik und E-Learning  
Universität Siegen  
Hölderlinstr. 3  
57068 Siegen  
stechert@die.informatik.uni-siegen.de

**Abstract:** Die Förderung des Informatiksystemverständnisses ist eine vorrangige Aufgabe des Informatikunterrichts. Die vorliegende Arbeit stellt eine Herangehensweise zur Entwicklung eines Unterrichtsmodells für die Lernenden in der Sekundarstufe II vor. Aufbauend auf einer Unterrichtsreihe zur objektorientierten Modellierung (OOM) wird ein kognitiver Zugang zu Wirkprinzipien in Informatiksystemen beschrieben. Dieser basiert auf objektorientierten Entwurfsmustern als Repräsentation vernetzter Fundamentaler Ideen der Informatik.

## 1 Motivation

Historisch wurden Computer als programmgesteuerte, isolierte Universalrechner betrachtet. Dies geht sowohl auf das theoretische Modell der Turingmaschine als auch auf die ersten elektronischen Rechenmaschinen zurück. Daraus resultierte jedoch eine Black-Box-Sichtweise auf den Einzelplatzrechner, dessen Eingabe- und Ausgabeverhalten durch sequenziell ablaufende Vorgänge erklärt wurde. Da es der Auftrag des Informatikunterrichts ist, das Verstehen und Gestalten der Lebenswelt zu fördern, wird die seit den 1990er Jahren zunehmend auftretende Vernetzung von Rechnern in lokalen Netzen und mit dem Internet für ein Informatiksystemverständnis relevant. Es „stellt sich ein Computersystem somit dar als Gruppe gleichrangiger, selbstständiger, einigermaßen intelligenter Akteure, die bestimmte Aufgaben erledigen und dazu miteinander und mit der Umgebung interagieren“ [BB95, S. 28]. Für den Informatikunterricht in der Sekundarstufe II rückt das für ein Systemverständnis notwendige vernetzte Denken in den Vordergrund: „Wissenschaftspropädeutisches Arbeiten erfordert problem- und prozessbezogenes Denken und Denken in Zusammenhängen“ [Ri99, S. XII]. Insbesondere „Fragen nach den Wirkprinzipien von Informatiksystemen sollen klären, wie diese aufgebaut sind, nach welchen Funktionsprinzipien ihre Komponenten effizient zusammenwirken und wie diese sich in größere Systemzusammenhänge einordnen lassen“ [Ri99, S. 6]. Wir werden im Folgenden den Begriff „Informatiksystem“ verwenden:

„Als Informatiksystem bezeichnet man die spezifische Zusammenstellung von Hardware, Software und Netzverbindungen zur Lösung eines Anwendungsproblems. Eingeschlossen sind alle durch die Einbettung des Systems in den Anwendungsbereich beabsichtigten oder verursachten nicht technischen Fragestellungen und ihre Lösungen, also Fragen der Gestaltung des Systems, der Qualifizierung der Nutzer, der Sicherheit sowie der Auswirkungen und Folgen des Einsatzes. Informatik ist dann die Wissenschaft von Entwurf und Gestaltung von Informatiksystemen.“ [CS03, S. 301]

Mit den Schwerpunkten Entwurf, Gestaltung und Vernetzung von Informatiklösungen ist der Rahmen gesteckt für ein Unterrichtsmodell zum Verstehen von Informatiksystemen und für ein tragfähiges kognitives Modell der Lernenden von Informatiksystemen. Grundlage dessen sind Beiträge zum Verständnis der spezifischen Zusammenstellung von Informatiksystemkomponenten. Zur Klärung des Begriffs „Unterrichtsmodell“ seien folgende Begriffsdefinitionen gegeben: Ein didaktisches Modell ist ein „Theoriegebäude zur Analyse und Modellierung“ von Lehr-Lern-Prozessen und klärt „theoretisch umfassend und praktisch folgenreich die Voraussetzungen, Möglichkeiten, Folgen und Grenzen“ didaktischen Handelns [JM03, S. 35]. Ein Unterrichtskonzept bietet „Gesamtorientierungen didaktisch-methodischen Handelns [...] Probleme didaktischer Theoriebildung treten oft [...] in den Hintergrund“ [JM03, S. 305f]. Auf dieser Basis sei ein Unterrichtsmodell definiert als ein theoretisch begründeter Rahmen für praktische Unterrichtskonzepte. Ein handlungsorientierter Lernprozess mit der OOM als Abstraktions-, Formalisierungs- und Entscheidungsprozess bietet sich als Basis für ein Unterrichtsmodell zur Förderung des Informatiksystemverständnisses mit Entwurfsmustern an. Denn eine ausgewogene Unterrichtsstruktur zu „Informatisches Modellieren“ und „Wirkprinzipien von Informatiksystemen“ ist für den Lernprozess anzustreben [SS04]. Dieser Artikel leistet einen Beitrag zur Bestimmung eines angemessenen Verhältnisses dieser Schwerpunkte indem Vorgehensweisen und Kriterien zur Ausgestaltung von Informatikunterricht hinsichtlich des Verstehens von Informatiksystemen entwickelt werden.

Zur Motivation des Lernprozesses stellt sich die Frage nach zeitgemäßen Konzepten und Modellen zur Förderung des Systemverständnisses bezüglich Aufbau, Arbeitsweise und Vernetzung, die in Abschnitt 2 beantwortet wird. Ausgangspunkt dieser Arbeit ist eine lernerzentrierte didaktische Klassifikation [St06] von objektorientierten Entwurfsmustern [Ga95] hinsichtlich ihres Beitrags zum Informatiksystemverständnis, deren Essenz ebenfalls in Abschnitt 2 vorgestellt wird. Die Vorgehensweise zur Erstellung eines Unterrichtsmodells für die angestrebten Bildungsziele, welches das informatische Modellieren und die Wirkprinzipien von universellen Informatiksystemen in das Zentrum des Lernprozesses stellt, wird in Abschnitt 3 erläutert. Der Begriff der Vernetzung wird auf vier Ebenen verwendet: Vernetzung von Informatiksystemen, Vernetzung von Entwurfsmustern, Vernetzung von Fundamentalen Ideen der Informatik nach Schwill [Sc93] und Vernetzung kognitiver Prozesse. Die Vernetzung soll helfen, die kognitive Brücke zwischen isolierten Fundamentalen Ideen der Informatik und dem Ziel des Verstehens von Informatiksystemen zu errichten. Im Fokus liegt dabei die durch Entwurfsmuster gegebene Vernetzung von Fundamentalen Ideen der Informatik, womit auch der Bildungswert begründet wird. Der vorliegende Artikel ist Teil des Dissertationsprojektes des Autors zum Informatiksystemverständnis [St05]. Die Forschungsvorgehensweise setzt sich aus fünf Phasen zusammen:

In der ersten, theoretisch-konstruierenden Phase werden wissenschaftliche Fragen aufgedeckt. In der zweiten Phase werden Hypothesen gewonnen. Dabei liegen geistige und praktische Aktivitäten der Lernenden im Fokus. In der dritten Phase wird die Schlüssigkeit der erarbeiteten Theorie und deren Stimmigkeit mit anerkannten Theorien der Fachwissenschaft, der Fachdidaktik und der Lerntheorie überprüft. Als vierte Phase schließt sich die Überprüfung im unterrichtlichen Geschehen mit einer Machbarkeits- und Akzeptanzstudie an. Schlussendlich erlaubt die Evaluationsphase die Rückkopplung auf die vorhergehenden Phasen. Die vorliegende Unterrichtsmodellentwicklung ist Bindeglied zwischen theoretischer Fundierung des Ansatzes und praktischer Umsetzung in Schulen und somit als Übergang von der dritten zur vierten Phase einzuordnen.

## **2 Informatiksystemverständnis**

### **2.1 Fachdidaktische Erkenntnisse zu Wirkprinzipien in Informatiksystemen**

Ziel des Forschungsvorhabens des Autors ist die Förderung des Informatiksystemverständnisses in der Sekundarstufe. Die OOM gilt als ein wichtiges und zeitgemäßes Konzept für den Informatikanfangsunterricht und wird vornehmlich in der Sekundarstufe II behandelt. Zur thematischen Eingrenzung des Forschungsvorhabens sei das Forschungsfeld auf den Informatikunterricht nach einem Anfangsunterricht zu OOM in der Sekundarstufe II beschränkt und die in Abschnitt 3 vorgestellte Vorgehensweise zur Erstellung eines Unterrichtsmodells darauf ausgerichtet. Hypothese ist, dass objektorientierte Entwurfsmuster als so genannte Wissensrepräsentationen – insbesondere für vernetzte fundamentale Ideen der Informatik – das Informatiksystemverständnis fördern. In Anlehnung an die Terminologie in der Kognitionswissenschaft [An01] und der Forschung zur künstlichen Intelligenz [CS03, S. 361] werden Wissensrepräsentationen verwendet, um zu komprimieren, zu veranschaulichen sowie das Lernen und Rekapitulieren zu unterstützen. Wissensrepräsentationen sind meist auf bestimmte Domänen beschränkt.

Die Hypothese ist begründet durch folgende Punkte:

1. Objektorientierte Entwurfsmuster sind als abstrahierte Lösungen wiederkehrender Entwurfsprobleme Wissensrepräsentationen für Entwurfsheuristiken zur Gestaltung von Informatiksystemen,
2. sie betonen den Vernetzungsaspekt, da sie als Strukturelemente auf vielfältige Weise kombiniert werden können („Pattern Language“),
3. ihnen sind fundamentale Ideen der Informatik inhärent, die – bei der Kombination von Entwurfsmustern – mit den fundamentalen Ideen der Informatik in anderen Entwurfsmustern vernetzt werden.

Demnach sind Entwurfsmuster Träger von fundamentalen Ideen der Informatik, d. h. sie sind eine Repräsentation für diese fundamentalen Ideen der Informatik. Damit ist insbesondere für den Informatikunterricht in der Sekundarstufe II der Bildungswert von Entwurfsmustern begründbar.

Außerhalb des hier Untersuchten ergibt sich die Frage nach weiteren Wissensrepräsentationen. Angedeutet sei ein Vergleich mit Algorithmen als Trägern Fundamentaler Ideen der Informatik. Algorithmen betonen die Masteridee „Algorithmisierung“ der Fundamentalen Ideen der Informatik sehr stark, während die beiden weiteren Masterideen „strukturierte Zerlegung“ und „Sprache“ meist nur unter dynamischen Gesichtspunkten eines ablaufenden Algorithmus transportiert werden. Im Gegensatz dazu können Entwurfsmuster immer auch als Modellierungsbausteine, d. h. der „strukturierten Zerlegung“ zugehörend, gesehen werden. Sie beschreiben aber ebenso oft dynamische Abläufe aus dem Bereich der „Algorithmisierung“ und der „Sprache“. Beispielhaft dafür seien die Entwurfsmuster Interpreter, zur Darstellung regulärer Sprachen, und Iterator, der die unterschiedlichen Traversierungsalgorithmen implementiert, genannt. Das schrittweise Analysieren der Komponenten des Informatiksystems, das Verbinden der Komponenten und Wirkprinzipien zu verzahnten Prozessen und das Problemlösen mit dem Informatiksystem stehen bei dem angestrebten Lernprozess zum Verstehen von Informatiksystemen im Vordergrund. Nach Ossimitz [Os00] umfasst systemisches Denken: Vernetztes Denken, Denken in Modellen, dynamisches Denken und systemisches Handeln. Zum vernetzten Denken sind entsprechende Darstellungsmittel notwendig. Das bedeutendste Denkwerkzeug der Systemwissenschaft ist das Wirkungsdiagramm [Os00]. Das Wirkungsdiagramm visualisiert die vernetzten Fundamentalen Ideen der Informatik.

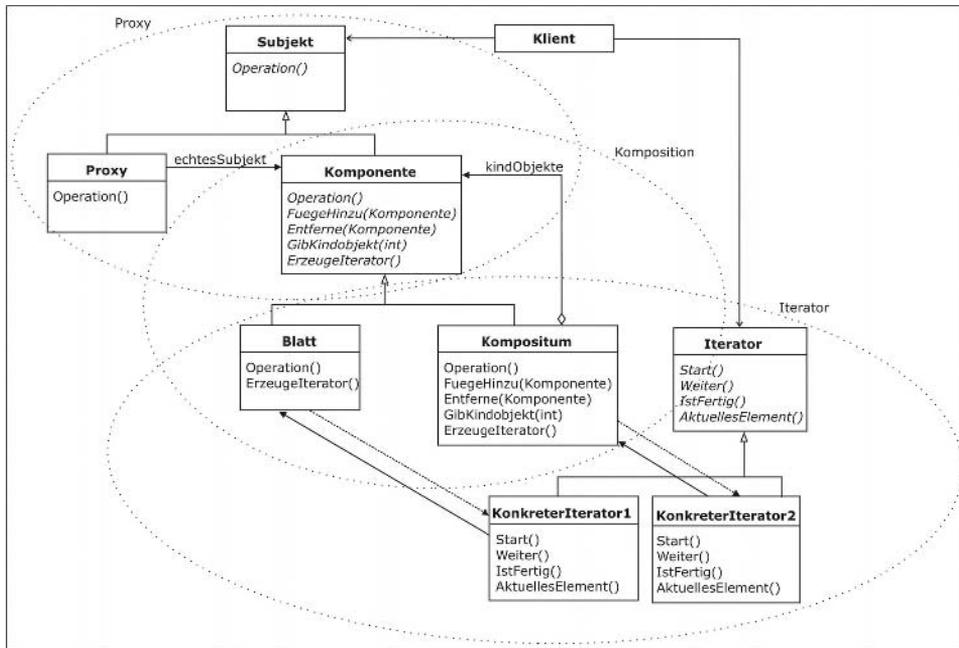


Abbildung 1: Klassendiagramm der kombinierten Entwurfsmuster

Das folgende in Abb. 1 dargestellte Beispiel veranschaulicht die Vernetzung von Entwurfsmustern und ihnen innewohnenden Fundamentalen Ideen. Es ist Grundlage des Unterrichtsmodells in Abschnitt 3, das über die für das Informatiksystemverständnis notwendige Vernetzung der Wirkprinzipien motiviert wird.

Die drei Entwurfsmuster Kompositum, Proxy und Iterator sind so kombiniert, dass der Iterator alle Elemente eines zusammengesetzten Objektes, des Kompositums, nacheinander iteriert.

Kompositum	Iterator	Proxy
Rekursion	Rekursion	Zugriffskontrolle
Baumstruktur	Polymorphie	Platzhalter/Referenz
Schnittstelle	In-/preorder Traversierung	Schnittstelle
Klassenhierarchie	Stack-Datenstruktur	Vererbung
Polymorphie	Schnittstelle	Interaktionsregelung
Teil-Ganzes-Beziehung		Kapselung
Vererbung		Polymorphie

Tabelle 1: Entwurfsmuster als Träger Fundamentaler Ideen der Informatik

Der Proxy kontrolliert den Zugriff auf das zusammengesetzte Objekt. Zur Auflockerung und Motivation der Lernenden können unterschiedliche Szenarios in Gruppenarbeit erstellt werden.

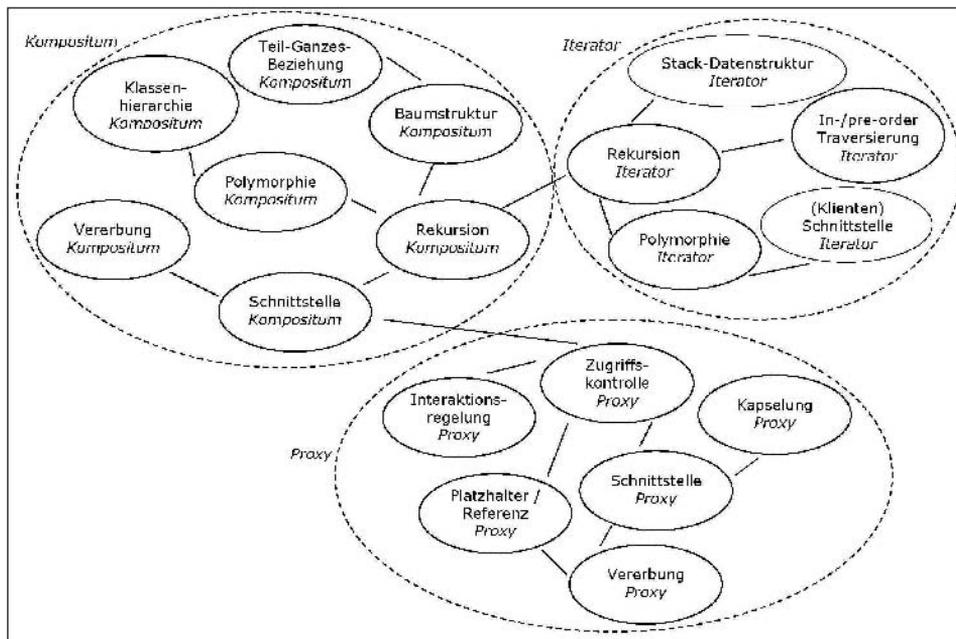


Abbildung 2: Wirkungsdiagramm zur Kombination der Entwurfsmuster

Diese können beispielsweise eine Speisekarte (Kompositum) mit Kellner (Proxy) und einem die Menüeinträge zählenden Gast (Iterator) sein oder eine zusammengesetzte Grafik (Kompositum), die in ein Dokument eingebunden wird.

Die Grafik wird durch einen leeren Rahmen (Proxy) bis zu einem konkreten Aufruf repräsentiert und anschließend von einem Leser auf die Anzahl der Objekte geprüft (Iterator). Einige der den Entwurfsmustern Kompositum, Proxy und Iterator zugehörigen Fundamentalen Ideen der Informatik sind in Tab. 1 bezüglich des Beispiels aufgeführt. Als Elemente des zu erstellenden Wirkungsdiagramms sind einige Fundamentale Ideen der Informatik mehrfach aufgeführt, entsprechend dem Entwurfsmuster, in dem sie enthalten sind. Die Reihenfolge der Fundamentalen Ideen der Informatik in Tab. 1 ist gewichtet nach Eignung des Entwurfsmusters als Repräsentation dieser Fundamentalen Idee der Informatik (vgl. [St06, HS02]). Die Vernetzung dieser Fundamentalen Ideen bei der Kombination der Entwurfsmuster ist in Abb. 2 ersichtlich. Die eingezeichneten Wirkbeziehungen sind nicht vollständig (vgl. [St06]), geben aber einen Überblick über wichtige Abhängigkeiten. Insbesondere die Verbindungspunkte der Entwurfsmuster sind hier beachtenswert. Die rekursive Struktur des Kompositums bedingt die Baumstruktur, die eine Teil-Ganzes-Beziehung beschreibt. Sie ermöglicht außerdem zusammen mit der Vererbung die einheitliche Schnittstelle des Kompositums und erzwingt eine rekursive Iteration. Zugriffskontrolle über den Proxy wird durch die einheitliche Schnittstelle des Kompositums ermöglicht. Auswahlkriterien für die genannten Wirkbeziehungen sind die Stufen des Lernprozesses, wie sie in Abschnitt 3 aufgeführt werden. Dadurch ergibt sich die Lesart für die Richtung der Wirkbeziehungen. Vernachlässigt wird in diesem Beispiel Sprache als Fundamentale Idee der Informatik, die durch die Darstellung in UML und die Kombination der Entwurfsmuster auf UML-Basis gegeben ist (vgl. [St06]). Die vernetzte Darstellung von Fundamentalen Ideen der Informatik in Wirkungsdiagrammen ist nicht auf OOM und Softwaresysteme beschränkt. Da die den Entwurfsmustern zugrunde liegenden Fundamentalen Ideen der Informatik Wirkprinzipien in allen Informatiksystemen sind, ist durch dieses Beispiel ein kognitiver Zugang zu größeren Systemen zu erwarten.

## 2.2 Fachwissenschaftliche Modelle zur Beschreibung von Informatiksystemen

Nach [CS03] lassen sich Informatiksysteme, d. h. die Zusammenfassung mehrerer Komponenten zu einer Einheit, auf folgende Kriterien untersuchen:

- ein nach außen sichtbares Verhalten (Black Box),
- eine innere Struktur (White Box),
- spezifikationsbedingte Eigenschaften (z. B. Implementierungsdetails).

Ein Ansatz zur Förderung des Verstehens von Informatiksystemen beruht auf Blockschaltbildern, die eine Synthese aus mathematisch-logischer Beschreibung und der technischen Umsetzung repräsentieren. Es wird jedoch nur die Vernetzung von Komponenten eines isolierten Informatiksystems beschrieben. Dynamische Aspekte werden nicht repräsentiert. Das Schichtenmodell kann als Weiterentwicklung des Blockschaltbildes eine hierarchische Strukturierung von Systemkomponenten darstellen und Komponenten gruppieren. Damit werden die für das Systemverständnis notwendigen Schnittstellen und der Kommunikationsfluss beschrieben. Automatenmodelle betonen die Zustände und Zustandsübergänge, also dynamische Aspekte.

Dies ermöglicht eine Beschreibung von Systemzuständen und Ereignisketten. In der OOM entspricht die Model-View-Controller-Architektur einer objektorientierten Variante des Schichtenmodells mit der Trennung von Datenhaltung, Fachkonzept und Darstellung. Zusammenfassend beschreiben Blockschaltbilder somit die Beziehungen zwischen Komponenten eines isolierten Informatiksystems, während 3-Schichtenmodelle als Architektur die Beziehungen zwischen Komponentengruppen abbilden. Die 4-Schichtenarchitektur des Internets beschreibt die Beziehungen zwischen Komponentengruppen und ist in direktem Zusammenhang mit der Vernetzung von Informatiksystemen zu sehen [PD03]. Die komponentenbasierte Sichtweise sei im Folgenden erweitert um die prozessbasierte Sichtweise. Hiernach werden die äußere Struktur eines Informatiksystems (Black-Box) und insbesondere dessen Architektur, durch große langsame Prozesse definiert, während das Innere des Systems durch kleine und schnelle Prozesse bestimmt ist. Diese neue Sichtweise wird notwendig, wenn wir Entwurfsmuster als Wissensrepräsentation von vernetzten Fundamentalen Ideen ansehen und sie als interagierende Prozesse einer Mikroebene im System untersuchen. Die Sichtweise als interagierende, sich wandelnde Prozesse verdeutlicht das für Systemverständnis notwendige vernetzte Denken. Entwurfsmuster fördern das Informatiksystemverständnis, da sie als Wissensrepräsentation für eine bestimmte Menge vernetzter Fundamentalener Ideen den Lernenden helfen, diese Ideen selbst wieder zu erkennen, einzusetzen und zu vernetzen. In nicht mit Entwurfsmustern erstellten objektorientierten Modellen ist das Erkennen solcher Wirkprinzipien ungleich schwieriger, da es keine wieder erkennbaren Strukturen gibt. In großen Softwareprojekten gestaltet sich jedoch selbst das Wiedererkennen einzelner Entwurfsmuster schwierig, da sie mit anderen Entwurfsmustern „verschmelzen“, d. h., dass sie ihre Klassenstruktur durch Synergieeffekte mit anderen Entwurfsmustern ändern. Hypothese ist hier, dass durch Kenntnis der Fundamentalener Ideen der Informatik in diesen Entwurfsmustern die Verschmelzung erklärbar und kognitiv leichter erfassbar wird.

### **3 Unterrichtsmodell für Informatiksystemverständnis**

Im Folgenden wird die Vorgehensweise zur Erstellung eines Unterrichtsmodells, d. h. eines theoretischen Rahmens für das Verstehen von Informatiksystemen im Informatikunterricht der Sekundarstufe II, vorgestellt. Grundlage des Modells ist eine lernerorientierte Klassifikation von einzelnen und zusammengesetzten Entwurfsmustern bezüglich der oben genannten Charakteristika von Informatiksystemen und den Fundamentalener Ideen der Informatik (vgl. [St06]). Der Lernprozess sei in Lernphasen strukturiert. Für jede Phase werden Lernziele, -inhalte, -methoden und ggf. Lernsoftware sowohl für die Mindestanforderungen als auch für darüber hinaus gehende Anforderungen angegeben. Die Vorkenntnisse der Lernenden in der Sekundarstufe II sind Basiskonzepte der OOM, die sie verstehen und anwenden können. Sie können Objekte identifizieren und beschreiben sowie deren Zustände und Eigenschaften erkennen. Formalisierte Abstraktion der Objekte zu Klassen mit Attributen und Operationen kann von den Lernenden durchgeführt werden. Die Beziehungen zwischen Klassen können als Assoziationen (hat-, kennt-Relationen und Vererbungshierarchien) beschrieben werden. Ausgewählte Elemente der Sprache „Unified Modelling Language“ (UML) können genutzt werden, um Klassen und Objekte zu interpretieren und selbstständig zu erstellen. Begründet durch den wissen-

schaftspropädeutischen Auftrag des Schulfachs Informatik in der Sekundarstufe II und der damit einhergehenden Notwendigkeit des Informatiksystemverständnisses, sind die Lernziele  $S_1$  des Unterrichtsmodells angelehnt an die drei Charakteristika eines zu analysierenden und zu konstruierenden Informatiksystems (vgl. Beispiel in Abschnitt 2.1):

- $S_1$ : Das Verstehen wesentlicher Aspekte des nach außen sichtbaren Verhaltens.
- $S_2$ : Das Verstehen von Aspekten der inneren Struktur, die auf Fundamentalen Ideen der Informatik als Wirkprinzipien basieren.
- $S_3$ : Das Verstehen spezifikationsbedingter Eigenschaften, die exemplarisch aus der Gruppe der typischen Eigenschaften ausgewählt werden.

Die Lernphasen werden spiralförmig wiederkehrend auf den größer werdenden Untersuchungsbereich des im Unterricht behandelten Informatiksystems angewendet. Nachfolgend werden diese Groblernziele zur Gliederung in drei Phasen genutzt. In der ersten Phase werden drei Teillernziele angestrebt:

- $S_{11}$ : Das Konzept der Liste und der Listentraversierung anhand des Black-box-Testens verstehen sowie weitere Anforderungen mittels Entwurfsmuster Iterator erklären können.
- $S_{12}$ : Das Konzept der Zugriffskontrolle anhand des Entwurfsmusters Proxy verstehen.
- $S_{13}$ : Phänomenologische Vorwegnahme des Konzepts der Baumstrukturen und deren Traversierung mittels Rekursion.

Mindestanforderungen in dieser Lernphase sind das Erkennen von Listen und Schlangen und die Notwendigkeit des schrittweisen Durchlaufens sowie der Zugriffskontrolle. Durch die Suche nach Lebensweltbeispielen wie der Warteschlange kann die Liste als Datenstruktur erkannt werden. Zur Traversierung der Liste kann nun eine dem Iterator ähnliche Struktur verwendet werden. Die Lernenden erarbeiten, wie ein Iteratorobjekt für eine konkrete Liste aussieht. Danach sollen sie ein Proxyobjekt für ein konkretes, vor Zugriff zu schützendes Subjekt objektorientiert modellieren. Es ist zu erwarten, dass das Konzept der Zugriffskontrolle durch den Lebensweltbezug schnell ersichtlich ist. Beispielsweise wird es im Internet durch den Proxy-Server und in der nicht technischen Lebenswelt durch einen Pförtner repräsentiert. Darüber hinaus können die Lernenden das Iteratorklassendiagramm erarbeiten, das von Traversierungsalgorithmus und Datenstruktur abstrahiert. Das Proxyklassendiagramm kann mit Blick auf die Fundamentalen Ideen der Informatik erarbeitet werden. So benötigt der Stellvertreter das von außen gleiche Aussehen (Schnittstelle) wie das zu vertretende Objekt (Vererbung, Polymorphie). Software, die den Lernprozess zur Iteration unterstützt, ist beispielsweise das Werkzeug ANIMAL zur Visualisierung von Algorithmen [Ro02].

In der zweiten Lernphase werden folgende Teillernziele angestrebt:

- $S_{21}$ : Das Konzept der Baumstruktur und der Rekursion anhand des Entwurfsmusters Kompositum verstehen.
- $S_{22}$ : Das Konzept der Traversierung von Baumstrukturen mittels Rekursion anhand der Entwurfsmusterkombination Kompositum und Iterator verstehen.

Mindestanforderung in dieser Lernphase ist das Erkennen von Hierarchieebenen, die Gruppierung von Elementen in Hierarchien, das Verstehen des Konzepts der Rekursion und das Traversieren von Baumstrukturen mit einem einfachen preorder-Algorithmus. Ein offener Unterrichtseinstieg zur rekursiven Zusammensetzung von Objekten aus Einzel- und zusammengesetzten Objekten lässt Schülerinnen und Schüler Lebensweltbeispiele nach Art des Entwurfsmusters Kompositums finden. Darüber hinaus können die Lernenden die Erweiterung der Baumstruktur um unterschiedliche Gruppierungen durchführen. Die Notwendigkeit der Terminierung und der Abbruchbedingungen wird von den Lernenden erkannt. Erarbeitung des Kompositumklassendiagramms durch die Lernenden sichert das bereits vorhandene Wissen zur OOM. Das Iterieren über die rekursive Struktur des Kompositums kann zusammen mit dem Prinzip des Stacks erarbeitet werden. Hierbei können verschiedene Algorithmen angewendet werden. Anschließend können die Lernenden ein konkretes Beispiel, wie die Schulhierarchie, selbst modellieren und programmieren.

In der dritten Lernphase wird folgendes Teillernziel angestrebt:

- S<sub>31</sub>: Das Konzept der Schnittstelle als Vertiefung des Vererbungsprinzips anhand der Kombination der Entwurfsmuster Kompositum und Proxy verstehen.

Mindestanforderung dieser Lernphase ist das Verständnis für die Kombination von Komponenten über definierte Schnittstellen. Darüber hinaus kann durch Kombination von Kompositum, Proxy und Iterator ein Lebensweltbeispiel modelliert werden (Speisekarte, Grafik; vgl. Beispiel in Abschnitt 2.1). Hilfreich kann eine Lernsoftware sein, wie sie zurzeit an der Universität Siegen von der studentischen Projektgruppe „Pattern Park“ erstellt wird [PP06]. In dieser Software werden verschiedene Entwurfsmuster vorgestellt, und die Kombination von Entwurfsmustern wird als Lerngegenstand behandelt. Alternativ kann zur Motivation ein vorgefertigtes Modell eines komplexen Lebensweltbeispiels bereits in der ersten Lernphase an die Lernenden ausgegeben werden.

## **4 Schlussfolgerungen und weitere Arbeit**

In diesem Beitrag wurde die Vorgehensweise zur Erstellung eines Unterrichtsmodells für Informatiksystemverständnis mit objektorientierten Entwurfsmustern als zeitgemäßer Repräsentation von vernetzten Fundamentalen Ideen der Informatik vorgestellt. Das Unterrichtsmodell beruht auf der Einbindung von verschiedenen Sichten auf ein Informatiksystem und der Vernetzung Fundamentaler Ideen der Informatik. Ein eingehender Vergleich mit weiteren Wissensrepräsentationen wie den Algorithmen ist durchzuführen. Zur Darstellung der Fundamentalen Ideen der Informatik in Entwurfsmustern wurde das Wirkungsdiagramm vorgeschlagen.

Nächster Schritt ist die Umsetzung des Modells in den Informatikunterricht der Sekundarstufe II. Die Gewichtung der Lernphasen ist mit Informatiklehrerinnen und -lehrern zu diskutieren. Nach der Erprobung inklusive einer Akzeptanzstudie wird das Unterrichtsmodell gemeinsam mit den Fachlehrerinnen und Fachlehrern evaluiert und verfeinert. Die Umsetzung des Unterrichtsmodells für Informatiksystemverständnis im Informatikunterricht der Sekundarstufe II, welches auf objektorientierten Entwurfsmustern basiert, wird ein zentraler Aspekt des vorgestellten Dissertationsprojekts sein.

## Literaturverzeichnis

- [An01] Anderson, J. R.: Kognitive Psychologie. Spektrum Akademischer Verlag, Heidelberg – Berlin, 3. Auflage, 2001.
- [BB95] Brauer W.; Brauer, U.: Informatik – das neue Paradigma. Änderungen von Forschungszielen und Denkgewohnheiten der Informatik. In: LOG IN, Nr. 4, 1995, S. 25-29
- [CS03] Claus, V.; Schwill, A.: Duden Informatik. Ein Fachlexikon für Studium und Praxis. Duden Verlag, Mannheim, 3. Auflage, 2003.
- [Ga95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA, 1995.
- [HS02] Harrer, A. and Schneider, M.: Didaktische Betrachtung zur Unterrichtung von Software-Mustern im Hochschulbereich. In Lecture Notes in Informatics, Volume 22, S. 67-76 (2002).
- [JM03] Jank, W.; Meyer, H.: Didaktische Modelle. Cornelsen Scriptor, Berlin, 6. Auflage, 2003.
- [Os00] Ossimitz, G.: Qualitative Systemwissenschaften I, Vorlesungsskriptum, Universität Klagenfurt. 2000, URL: <http://www.uni-klu.ac.at/~gossimitz/lv/usw00/skr/pdf/skriptum.zip> (21.02.2006).
- [PD03] Peterson, L.; Davie, B.: Computernetze. Eine systemorientierte Einführung. Dpunkt-Verlag, Heidelberg, 3. Auflage, 2003.
- [PP06] Pattern Park: Projektgruppe in der FG Didaktik der Informatik, Universität Siegen, URL: <http://www.die.informatik.uni-siegen.de/lehre/arbeiten/Projekt/projektgruppe> (24.06.2006).
- [Ri99] Ministerium für Schule und Weiterbildung, Wissenschaft und Forschung des Landes Nordrhein-Westfalen Richtlinien und Lehrpläne für die Sek. II – Gymnasium/Gesamtschule in Nordrhein-Westfalen: Informatik. Ritterbach Verlag GmbH, Frechen, 1. Auflage, 1999.
- [Ro02] Rößling, G.: ANIMAL-Farm: An Extensible Framework for Algorithm Visualization. Dissertationsschrift. Universität Siegen, 2002. URL: <http://elara.tk.informatik.tu-darmstadt.de/publications/2002/AnimalFarm.pdf> (24.07.2006).
- [Sc93] Schwill, A.: Fundamentale Ideen der Informatik. Zentralblatt für Didaktik der Mathematik 1 (1993) S. 20-31.
- [Sc05] Schubert, S.: From Didactic Systems to Educational Standards. In Samways, B. (ed.) 8th IFIP World Conference on Computers in Education (WCCE 2005); 4-7 July 2005.
- [SS04] Schubert, S.; Schwill, A.: Didaktik der Informatik. Spektrum Akademischer Verlag, Heidelberg, 2004.
- [St05] Stechert, P.: Entwurfsmuster in Bildungsprozessen. Doktorandenforum der 11. GI-Fachtagung Informatik und Schule - INFOS 2005, TU Dresden, URL: <http://www.die.informatik.uni-siegen.de/ruppe/stechert/publikationen/infos05.pdf> (24.07.2006).
- [St06] Stechert, P.: Informatics System Comprehension – A learner-centred cognitive approach to networked thinking. In CD Proceedings of IFIP TC3/WG3.1, WG3.3 & WG3.5 Joint Conference “Imagining the future for ICT and Education”. 26th-30th June 2006, Alesund, Norway, 2006.

## GI-Edition Lecture Notes in Informatics

- P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.
- P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.
- P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Language to Information Systems, NLDB'2001.
- P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.
- P-5 Andy Schürr (Hg.): OMER - Object-Oriented Modeling of Embedded Real-Time Systems.
- P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.
- P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods - Countering or Integrating the extremists, pUML'2001.
- P-8 Reinhard Keil-Slawik, Johannes Magenheimer (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.
- P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.
- P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.
- P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.
- P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.
- P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni (Hrsg.): The Changing Universities – The Role of Technology.
- P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.
- P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.
- P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.
- P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.
- P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.
- P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund.
- P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3.Okt. 2002 in Dortmund (Ergänzungsband).
- P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.
- P-22 Sigrid Schubert, Johannes Magenheimer, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.
- P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 - Fortschritt durch Beständigkeit
- P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

- P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003
- P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web
- P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin
- P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement - Erfahrungen und Visionen
- P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems
- P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications
- P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometric and Electronic Signatures
- P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003
- P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft
- P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)
- P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenber, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)
- P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenber (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik
- P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik
- P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003
- P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003
- P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004
- P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing
- P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste
- P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistence, Scalability, Transactions – Database Mechanisms for Mobile Applications
- P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services
- P-45 Bernhard Rumpe, Wolfgang Hesse (Hrsg.): Modellierung 2004
- P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment
- P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society
- P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications
- P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven
- P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

- P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm
- P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik
- P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004
- P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration
- P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Ad-hoc-Netze Medienintegration
- P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 - Informationssysteme im E-Business und E-Government
- P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen
- P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality
- P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments
- P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für informatische Bildung
- P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen
- P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit
- P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and its Applications
- P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005
- P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web
- P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik
- P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)
- P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)
- P-69 Robert Hirschfeld, Ryszard Kowalczyk, Andreas Polze, Matthias Weske (Hrsg.): NODE 2005, GSEM 2005
- P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)
- P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005
- P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment
- P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“
- P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology
- P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture
- P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz
- P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

- P-78 K.-O. Wenkel, P. Wagner, M. Morgens-  
tern, K. Luzi, P. Eisermann (Hrsg.): Land-  
und Ernährungswirtschaft im Wandel
- P-79 Bettina Biel, Matthias Book, Volker  
Gruhn (Hrsg.): Softwareengineering 2006
- P-80 Mareike Schoop, Christian Huemer,  
Michael Rebstock, Martin Bichler  
(Hrsg.): Service-Oriented Electronic  
Commerce
- P-81 Wolfgang Karl, Jürgen Becker, Karl-  
Erwin Großpietsch, Christian Hochberger,  
Erik Maehle (Hrsg.): ARCS'06
- P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.):  
Modellierung 2006
- P-83 Daniel Huson, Oliver Kohlbacher, Andrei  
Lupas, Kay Nieselt and Andreas Zell  
(eds.): German Conference on Bioinforma-  
tics
- P-84 Dimitris Karagiannis, Heinrich C. Mayr,  
(Hrsg.): Information Systems Technology  
and its Applications
- P-85 Witold Abramowicz, Heinrich C. Mayr,  
(Hrsg.): Business Information Systems
- P-86 Robert Krimmer (Ed.): Electronic Voting  
2006
- P-87 Max Mühlhäuser, Guido Rößling, Ralf  
Steinmetz (Hrsg.): DELFI 2006: 4. e-  
Learning Fachtagung Informatik
- P-88 Robert Hirschfeld, Andreas Polze,  
Ryszard Kowalczyk (Hrsg.): NODe 2006,  
GSEM 2006
- P-90 Joachim Schelp, Robert Winter, Ulrich  
Frank, Bodo Rieger, Klaus Turowski  
(Hrsg.): Integration, Informationslogistik  
und Architektur
- P-91 Henrik Stormer, Andreas Meier, Michael  
Schumacher (Eds.): European Conference  
on eHealth 2006
- P-93 Christian Hochberger, Rüdiger Liskowsky  
(Eds.): INFORMATIK 2006 – Informatik  
für Menschen, Band 1
- P-94 Christian Hochberger, Rüdiger Liskowsky  
(Eds.): INFORMATIK 2006 – Informatik  
für Menschen, Band 2
- P-95 Matthias Weske, Markus Nüttgens (Eds.):  
EMISA 2006: Methoden, Konzepte und  
Technologien für die Entwicklung von  
dienstbasierten Informationssystemen
- P-96 Saartje Brockmans, Jürgen Jung, York  
Sure (Eds.): Meta-Modelling and Ontolo-  
gies
- P-97 Oliver Göbel, Dirk Schadt, Sandra Frings,  
Hardo Hase, Detlef Günther, Jens Nedon  
(Eds.): IT-Incident Mangament & IT-  
Forensics – IMF 2006
- P-98 Hans Brandt-Pook, Werner Simonsmeier,  
Thorsten Spitta (Hrsg.): Beratung in der  
Softwareentwicklung Modelle - Metho-  
den - Best Practices
- P-99 Andreas Schwill, Carsten Schulte, Marco  
Thomas (Hrsg.): Didaktik der Informatik,  
3. Workshop

The titles can be purchased at:

Köllen Druck + Verlag GmbH  
Ernst-Robert-Curtius-Str. 14  
53117 Bonn  
Fax: +49 (0)228/9898222  
E-Mail: [druckverlag@koellen.de](mailto:druckverlag@koellen.de)