

Lehrrangements in der Informatiklehrerausbildung

Peter K. Antonitsch, Ulrike Lassernig, Andreas Söllei

Institut für Informatiksysteme/Informatik Fachdidaktik
Alpen-Adria Universität Klagenfurt
Universitätsstraße 65-67
A 9020 Klagenfurt
Peter.Antonitsch@uni-klu.ac.at
ulassern@edu.uni-klu.ac.at
asoellei@edu.uni-klu.ac.at

Abstract: Lehramtsstudien sollen auf die spätere Tätigkeit als Lehrer vorbereiten. Diese Vorbereitung darf sich nicht im Vermitteln fachlicher Kenntnisse erschöpfen, sondern muss auch das praktische Agieren des Lehrers berücksichtigen. Eine Möglichkeit dazu bieten Lehrrangements, die im universitären Umfeld Unterrichtssituationen nachbilden. Simulation von Informatikunterricht unter Verwendung informatischer Mikrowelten ist ein derartiges Lehrrangement, in dem die Studierenden sowohl Aspekte des Unterrichtens als auch grundlegende Konzepte der Informatik erfahren können.

1 Informatik und Unterrichten – Informatikunterricht

Das Lehramtsstudium Informatik an Universitäten hat das Ziel, Studierende auf ihre spätere Aufgabe, das Gestalten von Informatikunterricht vorzubereiten (vgl. z.B. [St04], S. 20ff). Dabei wird der informatische Blick primär in Fach-Lehrveranstaltungen vermittelt, die Sicht auf das Unterrichten im schulischen Kontext ist eher Domäne der pädagogischen Ausbildung. Die Vermittlung zwischen „Informatik“ und „Unterrichten“ obliegt der Informatik Fachdidaktik. Bedingt durch die Struktur universitärer Lehre erfolgt aber die didaktisch-methodische Aufbereitung informatischer Inhalte weitestgehend abgekoppelt von der Unterrichtswirklichkeit, d.h. praxisfern¹. Dies fällt umso stärker ins Gewicht, als der im Unterricht zu vermittelnde Inhalt ein wesentliches interaktives Moment des Lehrens und Lernens beinhaltet [Da05]: Zum einen sollen die Teilnehmer am Lernprozess über den Inhalt miteinander in Interaktion treten, zum anderen wird auch die Aneignung der Inhalte durch die innerhalb der Lerngruppe stattfindende Interaktion beeinflusst. Eine bloß theoretische Analyse von Unterricht kann dieser Dynamik aber kaum gerecht werden.

¹ Die (in Österreich) derzeit einzige Gelegenheit für Studierende, sich im Rahmen ihres Studiums praktisch mit ihrer zukünftigen Lehrerrolle im Informatikunterricht auseinanderzusetzen, besteht in einem zwölfwöchigen Schulpraktikum, vgl. [St04], S. 11.

Microteaching [A176] [Jü98] ist ein handlungsorientierter Ansatz zur Vorbereitung der Praxisphase an Schulen, der auch in der Informatiklehrausbildung angewandt wird². Komplexe Handlungsstrategien der Lehrperson werden in möglichst einfache Fertigkeiten zerlegt, die isoliert geübt, auf Video aufgezeichnet, reflektiert und (beliebig oft) wiederholt werden können. Das Üben und Reflektieren erfolgt in kleinen Gruppen von angehenden Lehrkräften, die wechselweise die Lehrer- und die Schülerrolle einnehmen. Ein derartiges „praxisnahes“ Setting im Rahmen universitärer Lehrveranstaltungen, in dem auch der Aspekt der Lehrer-Schüler Interaktion abgebildet werden kann, wird nachfolgend als Lehrarrangement bezeichnet. Microteaching berücksichtigt damit bis zu einem gewissen Grad die Forderung seitens systemisch-konstruktivistischer Pädagogik nach Selbsterfahrung von pädagogischen Beziehungen ([Re05], S. 268). Allerdings liegt der Fokus von Microteaching-Sequenzen primär auf der lehrerzentrierten Vermittlung von Inhalten und den sich daraus ergebenden Lehrer-Schüler Interaktionen. Dadurch ermöglichen sie nur begrenzt das Erleben der (und damit die Vorbereitung auf die) Komplexität des Unterrichts ([GG83] S. 20). Dieser Aspekt benötigt einerseits das Ineinandergreifen mehrerer Phasen von Unterricht und macht auch das Einbeziehen anderer, eher schülerzentrierter Sozialformen notwendig.

„Simulation von Unterrichtssituationen im universitären Umfeld“ [An06] versteht sich diesbezüglich als alternatives bzw. ergänzendes Lehrarrangement: Studierende des Lehramtes sollen im komplexitätsreduzierten Rahmen einer studentischen Lerngruppe eine längere Sequenz zu einem vorgegebenen Thema gestalten und reflektieren, sodass eigene Referenzerfahrungen für die spätere schulische Praxis ermöglicht werden. Im Rahmen des Schulpraktikums soll dann die Rückkoppelung mit diesen Referenzerfahrungen der Unterrichtssimulation erfolgen.

Hinsichtlich des Designs derartiger Unterrichtssimulationen für die Informatiklehrausbildung existieren allerdings zwei prinzipielle Problemfelder: Damit die Referenzerfahrungen mit Interaktionen in der Schulrealität vergleichbar werden, muss es einerseits gelingen, dass die Studierenden die Lehrer- und insbesondere die Schülerrolle möglichst echt annehmen. Andererseits muss ein Ansatz gewählt werden, der in der Simulation des Informatikunterrichts tatsächlich Interaktion zulässt, d.h. eher menschen- als computerzentriert ist.

2 Mikrowelten als Lehrarrangements

Mikrowelten (auch: „Mini-languages“) haben in der Informatik Fachdidaktik eine lange Tradition. Reduzierte und visualisierende Systeme wie Logo [Pa85], Karel der Roboter [Pa95] oder Kara der Marienkäfer [RNH04] werden als ideale Lernumgebungen für das Programmieren beschrieben und reflektiert (vgl. z.B. [Br97], [BOM99]). Auch der Transfer dieses Konzepts zur Verwendung von Tabellenkalkulationssoftware im Programmierunterricht wurde bereits angedacht [An05].

² Vgl. z.B. <http://ddi.cs.uni-potsdam.de/Lehre/SPS/> bzw. <http://ddi.cs.uni-potsdam.de/Lehre/SPS/MicroteachingBeispiele.htm> (25. 01. 2007).

Mikrowelten eignen sich aber auch als Basis für Lehrarrangements zur Durchführung von Unterrichtssimulationen in obigem Sinne. Erste diesbezügliche Erfahrungen gehen auf eine informatikdidaktische Lehrveranstaltung mit 10 Studierenden an der Alpen-Adria Universität Klagenfurt im Wintersemester 2004/05 zurück. Dabei waren von einem bzw. zwei der Studierenden als „Lehrer“³ mit Hilfe von Mikrowelten⁴ zwei simulierte Unterrichtseinheiten von ca. 50 Minuten Dauer zu gestalten. Die anderen sollten jeweils als „Schüler“ agieren, d.h. die gestellten informatischen Probleme lösen⁵.

Ausgehend von ihren subjektiven Theorien über (Informatik-) Unterricht hatten die „Lehrer“ den Unterricht eigenständig zu planen und durchzuführen und sollten über den „Stoff“ mit den „Schülern“ in Beziehung treten. Dabei zeigten sich angesichts der zuvor angesprochenen Problemfelder zwei wesentliche Vorteile der Verwendung von Mikrowelten:

- Face-to-face Interaktionen werden im Informatikunterricht durch das Werkzeug Computer eingeschränkt (siehe auch Abschnitt 3.2). Mikrowelten der betrachteten Art können aber leicht real nachgebaut werden⁶, die „Schüler“ (oder auch der „Lehrer“) können/müssen die Rolle des „Akteurs“ in dieser „Welt“ übernehmen. Dadurch wird einerseits das Miteinander-in-Beziehung-Treten wesentlich erleichtert, andererseits entsteht dadurch eine für die Studierenden ungewohnte informatische Perspektive: Nicht das individuelle Codieren sondern das gemeinsame (!) Problemlösen wird zur primären Tätigkeit des Programmierens (vgl. Abb. 1).



Abbildung 1: „Schülerin“ als „Akteur“ beim „Problemlösen“ in der Karel-Mikrowelt (links) bzw. „Lehrerin“-„Schüler“ Interaktion am realen Modell der Kara-Mikrowelt (rechts)

- Andererseits waren die Studierenden mit den Konzepten des Programmierens wohl vertraut, die verwendeten Mikrowelten waren aber unbekannt, sodass die als „Leh-

³ Im Sinne besserer Lesbarkeit wird bei Sammelbezeichnungen stets das grammatikalisch männliche Geschlecht verwendet. Dies meint aber selbstverständlich stets beide biologischen Geschlechter!

⁴ In diesen Unterrichtssimulationen wurden Kara der Marienkäfer bzw. eine Version von Karel dem Roboter verwendet.

⁵ Neben den beiden Co-Autoren Ulrike Lassernig und Andreas Söllei waren dies Ernst Aschbacher, Gudrun Egger, Michael Gyarmati, Irmgard Hausharter, Andreas Lukasser, Peter Moser, Rene Scheriau und Rainer Swatek. Ihnen sei an dieser Stelle für ihr engagiertes Mitwirken in den Unterrichtssimulationen und ihre kritischen Diskussionsbeiträge gedankt.

⁶ Dieser Aspekt der „methodischen Skalierbarkeit“ von Mikrowelten wurde bereits in [An05] beschrieben.

rer“ agierenden Studierenden aufgrund ihrer Unterrichtsvorbereitung einen Informations- bzw. Erfahrungsvorsprung erwerben konnten, der zur Herausbildung einer asymmetrischen (und daher schulähnlichen) „Lehrer“-„Schüler“ Beziehung ausreichte⁷.

In den durchgeführten Einheiten der Unterrichtssimulation folgte auf eine kurze Phase des „Lehrervortrages“, in dem die jeweils verwendete Mikrowelt erklärt wurde, das Lösen von vorgegebenen Problemstellungen in der Mikrowelt durch die „Schüler“. Lehrerzentrierter Unterricht und kooperatives Lernen, in dem die Lehrperson eine aktivierende bzw. beratende Funktion hat, lösten einander also ab. Dadurch konnten sich die Unterrichtssimulationen bis zu einem gewissen Grad als „Netz gegenseitiger Beziehungen (Erwartungen, Verhaltensweisen, Wahrnehmungen)“ entwickeln⁸, sodass trotz gründlicher Vorbereitung durch die „Lehrer“ stets ein Rest an Unsicherheit bei der Durchführung des „Unterrichts“ verblieb. Die Unterrichtssimulationen stellten so tatsächlich ein Modell für reales Unterrichtsgeschehen dar.

3 Erfahrungen

Die Unterrichtssimulationen wurden für die Reflexionsphase videographiert, um erste Anhaltspunkte über mögliche Lernfelder zu erhalten. In einer zweiten Phase wurden zwei der als „Lehrer“ agierenden Studierenden – Frau Lassernig und Herr Söllei – bei ihrem Schulpraktikum begleitet (und ihre „realen“ Unterrichtsauftritte ebenfalls auf Video aufgezeichnet), um Aufschluss darüber zu erhalten, inwieweit Referenzerfahrungen gemacht werden konnten. Die folgenden Ausführungen berücksichtigen die Ergebnisse beider dieser Phasen.

3.1 Unterrichten

Die Aufgabenstellung, nicht bloß Informationsinput in Form eines Lehrervortrages vorzubereiten, sondern eine gesamte „Unterrichtseinheit“ zu gestalten, verlangte von den „Lehrenden“, Unterricht als Prozess zu sehen. Zusätzlich zur inhaltlichen Planung mussten unter den gegebenen Rahmenbedingungen auch das Initiieren von Lernprozessen, die Organisation von Gruppenprozessen, das Interagieren mit den „Lernenden“ und das Abschließen der Einheit vorgedacht werden. Dadurch wurden Erfahrungen nicht nur im Hinblick auf das informatische Thema, sondern auch auf das Unterrichten (vor dem Hintergrund eines informatischen Themas!) ermöglicht:

Ein wesentlicher Referenzbereich war die für die Studierenden geänderte Wahrnehmung der Lehrerrolle. In der Reflexionsphase getätigte Äußerungen wie: „(...) man kennt eben nur seine eigene Schulzeit (...) wo man als Schüler nur die Durchführung des Unterrichts wahrnimmt]“ bzw. „vorher (...) die Lehrerrolle einfacher gesehen: Tests, Hausübung geben, prüfen (...) nachher: Lehrer in unterschiedlichen Rollen [als Leiter und als Begleiter/Berater]“ weisen auf Referenzerfahrungen bezüglich der tatsächlichen Dimensionen

⁷ Vgl. [An06] zum zu Grunde gelegten Bild der Lehrer- und Schülerrolle.

⁸ Vgl. [AP98], S. 78 zur systembezogenen Sichtweise von Unterricht.

von Lehrerhandeln im Unterricht hin. Besondere Beachtung fand dabei die gegenseitige Beeinflussung von Unterrichtsplanung und Unterrichtsdurchführung, entsprechend der Planung von Lernarrangements und den intendierten Interaktionen.

Ein zweiter Referenzbereich betraf die Wahrnehmung der Schüler durch den Lehrer. In Unterrichtssimulationen kennen die „Lehrer“ die „Schüler“ auch als Kollegen, d.h. in anderem sozialen Kontext, wissen daher um deren (in diesem Fall: informatisches) Können. Eine Referenzwahrnehmung ergab sich aus unterschiedlichem Verhalten zweier als „Schüler“ agierende Studienkollegen, die beide als „ausgezeichnete Informatiker“ geschätzt wurden, von denen der eine aber sehr zurückhaltend, der andere hingegen aktiv an der Unterrichtssimulation teilnahm. Das Beobachten der Schüler im Unterricht allein ließe offenbar nur bedingt Schlüsse auf deren Können zu, so die Schlussfolgerung der Studierenden. Vielmehr bedürfe es auch der verstärkten Auseinandersetzung mit dem individuellen Lernhintergrund von „auffälligen“ Schülern. Dafür wurde in der Reflexionsphase der Begriff der „Subjekt-sensiblen Didaktik“ geprägt.

Derartige Erfahrungen und deren Reflexion zeigten den im Rahmen der Unterrichtssimulationen „Lehrenden“ die Bedeutung der Beziehungsebene auch für den Informatikunterricht. Durch den Kontrast zu den ursprünglich vorhandenen subjektiven Theorien über Unterricht gelang damit zusätzlich eine erste Orientierung hin zu einem „beruflichen Selbstbild“ zwischen den Polen des „Fachexperten“, des „didaktischen Experten“ und des „pädagogischen Experten“ [BVV00].

3.2 Informatik

Die aktive Interaktion/Kommunikation⁹ zwischen Lehrern und Schülern ist Grundlage jedes Unterrichts [AT06]. Trotzdem fehlt dieser Aspekt im vorangegangenen Abschnitt, obwohl das Erzielen möglichst realitätsnaher Interaktion ein Hauptanliegen beim Design der Unterrichtssimulationen war. Aus Sicht der Autoren ist aber Kommunikation im Informatikunterricht spezifisch von den Gegebenheiten im Informatikunterricht bestimmt, damit aus informatischem Blickwinkel zu betrachten.

Informatikunterricht in der Schule findet häufig (üblicherweise?) in Computerräumen statt. Aufgrund ergonomischer Überlegungen (Lichteinfall, gerade Sitzhaltung,...), und um die Aufmerksamkeit der Schüler auf den Vortragenden/die Tafel/den Beamer zu fokussieren, werden dort Computer meist so positioniert, dass sie als „natürliche“ Barriere zwischen Lehrer und Schüler bzw. auch zwischen den in verschiedenen Reihen sitzenden Schülern stehen (vgl. Abb. 2a).

In diesem Punkt ergab sich eine Differenz zwischen der Schulwirklichkeit und den Unterrichtssimulationen. Letztere wurden ja über weite Strecken ohne Computer durchgeführt, die Studierenden konnten als Akteure in einer Mikrowelt die gestellten Probleme durch Darstellen und Diskutieren – eben kommunikativ – lösen. Gerade aus diesem Kontrast zwischen Kommunikation in einer face-to-face-Situation und der Kommunikationssituation „am Computer“ erwuchs eine weitere Referenzerfahrung:

⁹ Die Begriffe Kommunikation und Interaktion werden hier synonym verwendet.



Abbildung 2 a: Struktur eines „typischen“ Computerraums mit Computern als „Barriere“
 Abbildung 2 b: Beispiel für „indirekte Mensch-Computer-Mensch Kommunikation“

Bei einer Gesprächssituation, in der von einer Schülerin Hilfe bei der Lösung eines Codierungsproblems erbeten wurde, wurde die Kommunikation über den Computerbildschirm vermittelt: Die Frage (Aussage!) der Schülerin: „Das da [zeigt und blickt auf den Bildschirm] funktioniert nicht (...)“ wurde von der Praktikantin („Lehrerin“) mit „Da musst Du diese beiden Werte [zeigt und blickt auf den Bildschirm] verknüpfen (...)“ beantwortet und „gelöst“ (vgl. Abb. 2b). Obwohl sich also die beiden Kommunikationspartner direkt nebeneinander befanden, lief die Kommunikation ohne direkten Blickkontakt ab. Wir bezeichnen dieses Handlungsmuster¹⁰ im Informatikunterricht als „indirekte Mensch-Computer-Mensch Kommunikation“.

Was bedeutet dies für das Lernen von Informatik? Eine von der Schülerin offenbar als problemhaltig erkannte Situation wurde anhand der vorhandenen Problemrepräsentation am Bildschirm „gemeinsam mit der Lehrerin bearbeitet“, die Darstellung des Problems dabei aber nicht verändert. Dies unterschied sich deutlich von den Prozessen der Problembearbeitung im Rahmen der Unterrichtssimulationen, wo das Umformulieren des Problems stets Teil der Problemlösung war. Aufgabe des Lehrers im Informatikunterricht müsse es daher sein, die Schüler auch zur jeweils individuellen Darstellung eines wahrgenommenen Problems anzuleiten, so eine Überlegung bei der Reflexion dieser beiden Situationen¹¹. Dadurch ergab sich aus einer vom Standpunkt der Kommunikation problematisch erlebten Situation eine Referenzerfahrung für didaktisches Handeln im Informatikunterricht.

Problemrepräsentation ist im Zusammenhang mit dem (Erlernen von) Programmieren allerdings aus zwei unterschiedlichen Blickwinkeln zu sehen: Einerseits wird Programmieren häufig (auch von Informatikstudierenden) mit dem Erlernen einer Programmiersprache bzw. dem Codieren des „fertigen Programms“ gleichgesetzt. Die Darstellung des Problems und dessen Aufbereitung für die maschinelle Lösung durch den Computer wird demnach als dem Programmieren vorgelagerte Tätigkeit verstanden.

¹⁰ Dabei ist anzumerken, dass dieses Handlungsmuster auch bei „erfahrenen“ Informatiklehrern immer wieder zu beobachten ist (jeder beobachtet sich in derartigen Situationen selbst!).

¹¹ Vgl. z.B. [PG90], S. 48 oder [Br94], S. 204 zum Zusammenhang zwischen Problemlösekompetenz und der Fähigkeit, das Problem „passend“ darzustellen)

Durch das Fehlen von Computerarbeitsplätzen in der Unterrichtssimulation war dort aber gerade diese Tätigkeit der wesentliche Inhalt der Programmiererfahrung, sowohl für „Lehrer“ als auch für „Schüler“. Dadurch wurden die Studierenden sensibilisiert, den Begriff des Programmierens zumindest unter didaktischen Erwägungen weiter zu fassen und als durchgängigen Problemlösungsprozess zu verstehen. Das Codieren als „Überantworten der Lösung an den Computer“ wurde damit zum Test, ob die Problemlösung „richtig“ ist.

Dies weist auf den zweiten Aspekt der Problemrepräsentation beim Programmieren hin: Die Darstellung des Problems orientiert sich dabei immer an den Möglichkeiten der verwendeten Software. Im Kontext von Mikrowelten sind dies die Fähigkeiten des jeweiligen Akteurs und die Beschränkungen, die ihm durch seine „Lebenswelt“ auferlegt werden¹². Diese Sichtweise findet sich bei Duchâteau [Du92], der Programmieren definiert als das Delegieren einer Aufgabe¹³ an einen Ausführenden/Akteur unter Mitgabe einer Gebrauchsanweisung (Programm), die ihn zum Bewältigen der Aufgabe befähigt. Diese Sichtweise zeigte sich auch implizit in den Problemlösungen der Studierenden, z.B. als in der verwendeten Karel-Roboterwelt die Fähigkeiten dadurch eingeschränkt waren, dass nur Linksdrehungen, nicht jedoch Rechtsdrehungen möglich waren. Dadurch wurde eine (wenn auch nur geringfügige) Änderung der ursprünglichen Problemaufbereitung notwendig. „Implizit“ meint dabei, dass wohl die Änderung bewusst durchgeführt (und auch verbalisiert) wurde, dass die diesbezügliche Bedeutung der Software-Umgebung aber erst in der Reflexion gesehen wurde.

4 Resümee und Ausblick

In [An06] wurde darauf hingewiesen, dass sich das Konzept der Unterrichtssimulation als Lehrarrangement nicht nur mit informatischen Mikrowelten, sondern auch mit anderen, methodisch adaptierbaren Themen umsetzen lässt. Die methodische Adaptierbarkeit ermöglicht dabei in der beschriebenen Weise das Annehmen der Lehrer- bzw. der Schülerrolle und so das Entstehen von realitätsnahen Interaktionen im Rahmen der Unterrichtssimulation. Gleichzeitig können durch die methodische Adaption etablierter Themen auch Kontrasterfahrungen zur Situation im realen Schulunterricht hergestellt werden. Sowohl Situationen, die direkt auf die Schulpraxis übertragen werden können, als auch solche, die im Kontrast zur gängigen Praxis stehen, scheinen nämlich für das Herausbilden von Referenzerfahrungen wichtig zu sein.

Vom Standpunkt der Informatikdidaktik wird mit der Forderung nach methodischer Adaptierbarkeit zwar die Auswahl von Themen für die Simulation von Informatikunterricht eingeschränkt, Unterrichtsentwürfe nach dem Muster der Informatik-Werkstätten¹⁴ zeigen aber auch hier Möglichkeiten auf, Unterrichtssimulationen im universitären Umfeld mit einer anderen als der vorgestellten thematischen Ausrichtung umzusetzen. Den-

¹² Aufgrund dieser Sichtweise wird in diesem Artikel auch der Begriff „Mikrowelt“ und nicht „Mini-Sprachen“ verwendet. Die Sprache definiert einen wesentlichen Teil der Umgebung des Akteurs, seine „kleinen Welt“.

¹³ Duchâteau spricht von Aufgaben anstelle von Problemen und argumentiert, dass speziell im Anfangsunterricht die kleinen Herausforderungen die Bezeichnung „Problem“ wohl nicht verdienen.

¹⁴ Vgl. z.B. <http://www.swisseduc.ch/informatik/werkstatt/> (18. 04. 2007)

noch bleibt auch das Programmieren in diesem Zusammenhang interessant. Die geschilderten Ergebnisse deuten darauf hin, dass durch die Erfahrungen im Rahmen der durchgeführten Unterrichtssimulationen Studierende zum Nachdenken über den „didaktischen Kern“ des Programmierens angeleitet werden können: Ein Programm bewirkt, dass „etwas“ in einer bestimmten „Arbeitsumgebung“ eine Aufgabe löst.

Im Fall von Mikrowelten ist diese Arbeitsumgebung eine „conceptually simple notional machine“ [BOM99], im Fall professioneller Programmiersprachen (die dennoch in Schulen zum Erlernen des Programmierens verwendet werden¹⁵) die jeweilige virtuelle Maschine bzw. der Computer selbst. Damit erfordert aber das Arbeiten mit einer derartigen Programmierumgebung – wie der Name schon sagt – auch das Kennen und (einfach) Beschreiben-Können der Umgebung. Die Umgebung eines Programms ist – in letzter Instanz – die Hardware inklusive der durch die Programmiersprache auferlegten Beschränkungen. Die zugehörige „conceptually simple notional machine“ kann das Modell einer Register- oder Turingmaschine sein. Programmieren bedeutet somit mehr als „nur“ Codieren, mehr als „nur“ Problemlösen. Programmieren ist ganz im Sinne von [De03] „ein Fenster in die Welt der Computational Mechanics“ und verknüpft Bereiche des Informatikstudiums, die von Studierenden nicht selbstverständlich als zusammengehörig erkannt werden.

Zwar konnte dieser „Transferschritt“ von „Mikrowelten“ zur „Computerwelt“ beim ersten Durchgang von Unterrichtssimulationen (noch) nicht gemacht werden, die Erfahrungen der Studierenden und mit den Studierenden deuten aber darauf hin, dass derartige Lehrarrangements bei geeigneter Begleitung die skizzierte Sichtweise bei zukünftigen Informatiklehrern fördern können. Dass dabei zusätzlich individuell wertvolle (Vor-) Erfahrungen über das Unterrichten gemacht werden (können), ist für die Ausbildung von Lehrern der Informatik wohl mehr als nur beiläufiges Beiwerk.

Literaturverzeichnis

- [Al76] Allen D. (1976): Microteaching – ein Überblick. In: (Zifreund W. Hrsg.): Training des Lehrverhaltens und Interaktionsanalyse. Beltz, Weinheim und Basel 1976; S. 1-28
- [An05] Antonitsch P.K.: Standard Software as Microworld?. In (Mittermeir R.T. Ed.): From Computer Literacy to Informatics Fundamentals. Proc. 1st Int. Conf. On Informatics in Secondary Schools – Evolution and Perspectives (ISSEP), Klagenfurt 2005. Springer, Berlin Heidelberg, 2005; S. 189-197
- [An06] Antonitsch P.K.: Simulation von Unterrichtssituationen im universitären Umfeld als Beitrag zum Erwerb von Praxiskompetenz. In (Hilligus A.H., Rinkens H.-D. Hrsg.): Standards und Kompetenzen – neue Qualität in der Lehrerbildung? LIT Verlag, Berlin 2006; S. 391-397
- [AP98] Altrichter H., Posch P.: Lehrer erforschen ihren Unterricht. 3. Aufl. Klinkhardt, Bad Heilbrunn 1998
- [AT06] Antonitsch P.K., Tischer K.: Kommunikation im sozialen System Schule. In (Antonitsch P., Delanoy W., Palencsar F., Theuermann A., Tischler K. Hrsg.): Ich + Du ≠ Wir. Wege zur Kommunikation in der LehrerInnenbildung. dravaDiskurs, Klagenfurt 2006; S. 15-28

¹⁵ Die verwendeten Programmiersprachen überdecken dabei – zumindest in Österreich – ein weites Spektrum von der Objektorientierung (Java, C#) über Skriptsprachen bis hin zu „maschinennahen“ Sprachen wie C.

- [Br94] Brusilovsky P.: Explanatory Visualization in an Educational Programming Environment: Connecting Examples with General Knowledge. In: (Blumenthal B., Gornostaev J., Unger C. Eds.): *Human-Computer Interaction. Selected Papers of the 4th International Conference, EWHCI '94*, St. Petersburg 1994. Springer, Berlin Heidelberg 1994, S. 202-212
- [Br97] Brusilovsky P., Calabrese E., Hvorecky J., Kouchnirenko A., Miller P.: Mini-languages: A Way to Learn Programming Principles. In: *Education and Information Technologies 2* (1) 1997; S. 65-83
- [BOM99] du Boulay B., O'Shea T., Monk J.: The black box inside the glass box: presenting computing concepts to novices. In: *International Journal of Human-Computer Studies*, 51(2) 1999; S. 265-277
- [BVV00] Beijaard D., Verloop N., Vermunt J.D.: Teacher's perceptions of professional identity: an exploratory study from a personal knowledge perspective. In: *Teaching and Teacher Education*, 16 2000, S. 749-764
- [Da05] Dauber H.: Der Lehrer in der Schule der Zukunft: Coach oder Pädagoge? In: (Dauber H., Krause-Vilmar D. Hrsg.): *Schulpraktikum vorbereiten. Pädagogische Perspektiven für die Lehrerbildung*. Klinkhardt, Bad Heilbrunn 2005, S. 23-37)
- [De03] Denning P.J.: Great Principles of Computing. In: *Communications of the ACM* Vol. 46, No. 11 November 2003
- [Du92] Duchâteau C.: From "DOING IT..." to "HAVING IT DONE BY...": the heart of programming. Some didactical thoughts. In: *Preproceedings NATO ARW Cognitive Models and Intelligent Environments for Learning Programming*. S. Margherita Ligure, Genova 1992
- [GG83] Grell J., Grell M.: *Unterrichtsrezepte*. Beltz, Weinheim und Basel 1983
- [Jü98] Jürgens B.: Wie lernen Lehrer Lehrerverhalten? In: *Praxis Schule* 5–10, Heft 4/1998, S. 38-43
- [Pa85] Papert S.: *Gedankenblitze. Kinder, Computer und Neues Lernen*. Rororo, Reinbek bei Hamburg 1985
- [Pa95] Pattis R.E.: *Karel the Robot. A Gentle Introduction to the Art of Programming*. 2nd ed. John Wiley & Sons, New York 1995
- [PG90] Pennington N., Grabowski B.: The Tasks of Programming. In: (Hoc J.-M., Green T.R.G., Samurçai R., Gilmore D.J. Eds.): *Psychology of Programming*. Academic Press, London 1990, S. 45-62
- [Re05] Reich K.: *Systemisch-konstruktivistische Pädagogik*. Beltz, Weinheim und Basel, 5. Aufl. 2005
- [RNH04] Reichert R., Nievergelt J., Hartmann W.: *Programmieren mit Kara. Ein spielerischer Zugang zur Informatik*. Springer, Berlin Heidelberg 2004
- [St04] Studienplan Lehramt an der Fakultät für Wirtschaftswissenschaften und Informatik an der Universität Klagenfurt 2004; verfügbar unter: <http://www.uni-klu.ac.at/home/stplaene/wiinfo/lawiinfo04.pdf> (25. 01. 2007)

