

# Einführung in visuelle Programmiersprachen und Mobile Endgeräte

Büdding, Hendrik

Institut Didaktik der Mathematik und Informatik  
Fachbereich 10, Westfälische Wilhelms-Universität  
info@m-learning.info

Wie lassen sich kerninformatische Inhalte im Informatikunterricht mit mobilen allgegenwärtigen Endgeräten vermitteln? Was ist der Mehrwert? Döring und Kleeberg [DK06] berichten über Mobiles Lernen in verschiedenen Projekten und beschreiben die technischen und didaktischen Aspekte des mobilen Arbeitens mit portablen Endgeräten im schulischen Umfeld. Aufgrund der positiven Bilanz, die dieser Text zulässt, aber auch aufgrund anderer Ergebnisse, beispielsweise des „Palm Education Pioneers Program“ bzw. Arbeiten von P. Haller („PDA macht Schule“), scheint die Entscheidung, Mobile Endgeräte für den Informatikunterricht zu nutzen, als grundlegend sinnvoll. Im Rahmen eines fachdidaktischen Ansatzes, der die Entwicklung des Ubiquitous und Pervasive Computing für den Informatikunterricht in der Schule anhand von Handheld Computern [ML06] (in der Hand gehaltene Computer / PDA) mit einer entsprechenden Schulungssoftware aufgreift, wird der Informatikunterricht um weitere Aspekte und Unterrichtsmethoden bereichert. Erfahrungsberichte zur Integration von PDAs in die Informatik-Ausbildung findet man z.B. von Luckin, Brewster et al bei Kukulska-Hume und Traxler.<sup>1</sup> Kukulska-Hume und Traxler beschreiben das Mobile Lernen als spontan, persönlich, tragbar und situativ; aber es kann auch informell, unaufdringlich, allgegenwärtig und durchschlagend passieren.<sup>2</sup> Um einen real existierenden Prozess, der bislang unvollständig oder falsch wahrgenommen wurde und dessen mentale Fehlvorstellung der Schülerinnen und Schüler (SuS) existiert, richtig zu begreifen, müssen auch mobile Lernszenarien berücksichtigt werden, die außerhalb des schulischen Computerraums stattfinden. Dabei liegt die Bedeutung der mobilen Programmentwicklung darin, innerhalb der Modellierungsphase ein möglichst genaues, abstrahiertes Modell abzubilden und den entwickelten Algorithmus anhand „realer Situationen“ zu erproben, d.h. mit realen Prozessen direkt zu vergleichen und somit gesichert zu verifizieren. Ebenso ist die VOR-ORT Programmentwicklung und -verifikation durch den Besuch des späteren realen Einsatzgebiets der IT vorteilhaft, da die SuS darin einen verstärkten Realitätsbezug bei ihrer Programmentwicklung erkennen. Erste Ergebnisse unter Verwendung der Entwicklungsumgebung „PocketCoder“ [PC06] (erprobt mit dem Fujitsu Siemens Pocket LOOX 720) stimmen optimistisch. Vor-Ort-Programmierung scheint den Modellierungsprozess zu unterstützen. Es gelingt besser, die grundsätzliche Lösungsidee für ein algorithmisches Problem zu finden und abzubilden als in einer stationären Situation. In der Praxisphase des hier vorgestellten Projektes wurde und wird untersucht, ob es möglich ist, traditionelle Inhalte des Informatikunterrichts mit Handheld PDAs zu vermitteln. Neben den technischen Möglichkeiten mobiler Computer sollen insbesondere Auswirkungen auf die inhaltliche und methodische Gestaltung des Unterrichts analysiert werden. Visuelle Programmiersprachen können den Einstieg in die Programmierung erleich-

---

<sup>1</sup> Vgl.: [KH05] S. 116-123.

<sup>2</sup> Vgl.: [KH05] S. 25-44.

tern. Zwar bietet laut Schiffer [Sc98] die visuelle Programmierung im Regelfall eine eingeschränktere Ausdrucksweise als ein textuelles System, aber in der Einstiegsphase im Informatikunterricht hat ein visuelles Programmiersystem den Vorteil, dass sich die Fehlerarten und -zahlen reduzieren lassen. Die SuS sollen die Syntax verstehen, die der Programmiersprache zugrunde liegt, sollen sich aber hauptsächlich mit der Modellierung und deren Implementierung in die Programmiersprache beschäftigen, um die globalen Programmierparadigmen/-konzepte zu verstehen. Eine „interaktive“ visuelle Programmiersprache, wie PocketCoder, zeigt automatisch Alternativen an, die syntaktisch an der gerade bearbeiteten Stelle zulässig sind. Durch den speziellen Aufbau der visuellen Programmiersprache, die nur eingeschränkte, syntaktisch korrekte Auswahlmöglichkeiten zulässt, sind die SuS in der Lage, schneller zu lernen und erhalten aufgrund der geringeren Fehlerquote bei der Codeerstellung ein schnelleres Erfolgserlebnis (weitere Beispiele in [Th99]). PocketCoder wurde am Arbeitsbereich Didaktik der Informatik an der Universität Münster entwickelt und basiert auf Erkenntnissen einer Bedarfsbefragung von Informatiklehrern in NRW. Das System besitzt eine ikonische Programmierumgebung, bei der mit Hilfe einer "Klicken und Auswählen"-Bedienoberfläche ein an Python angelehntes Programm erstellt werden kann. Die Idee ist, PocketCoder in den Anfangsunterricht zu integrieren, um SuS an die allgemeinen und pythonspezifischen Programmierkonzepte heranzuführen und den Brückenschlag zu komplexen Python-Entwicklungen im Informatikunterricht ggf. auf Desktop-PC zu erleichtern bzw. vorzubereiten. Die Programmiersprache ermöglicht es, mehreren PDAs bzw. Pockt-PCs unterschiedliche Aufgaben zuzuordnen und diese im Rahmen der Verteilten Programmierung miteinander über WiFi arbeiten zu lassen.

## Literaturverzeichnis

- [DK06] Döring, N.; Kleeberg, N.: Mobiles Lernen in der Schule. Entwicklungs- und Forschungsstand. Unterrichtswissenschaft - Zeitschrift für Lernforschung, 34 (1), S. 70-92, 2006.
- [Hu06] Humbert, L.: „Stifte und Mäuse“ auf mobilen Systemen. In: Humbert, L.: If Fase (2006), Ausgabe 8, 1. April 2006;
- [KH05] Kukulska-Hulme, A. et al: Mobile learning : a handbook for educators and trainers, London, 2005.
- [ML06] M-Learning Projekt Handhelds@School, <http://www.m-learning.info>, 01.04.2007.
- [PC06] PocketCoder – Projekt Universität Münster, [www.pocketcoder.de](http://www.pocketcoder.de), 01.04.2007.
- [Sc98] Schiffer, S.: Visuelle Programmierung – Grundlagen und Einsatzmöglichkeiten. Addison-Wesley-Longman, Bonn, 1998.
- [Th99] Thomas, M.: <http://ddi.uni-muenster.de/personen/marco/visu/FolienVP3/index.htm> (31.01.2007).